DOCUMENT RESUME

ED 108 513 FL 006 984

AUTHOR	Wang, William S-Y; And Others
TITLE	Chinese-English Machine Translation System.
INSTITUTION	California Univ., Berkeley.
SPONS AGENCY	Rome Air Development Center, Griffiss AFB, N.Y.
REPORT NO	RADC-TR-75-109
PUB DATE	Apr 75
NOTE	219p.
AVAILABLE FROM	National Technical Information Service, Springfield, Virginia 22151
EDRS PRICE	MF-$0.76 HC-$10.78 PLUS POSTAGE
DESCRIPTORS	*Chinese; *Computational Linguistics; *Computer Programs; Contrastive Linguistics; Data Processing; Descriptive Linguistics; Information Processing; Lexicography; *Machine Translation; Phrase Structure; Semantics; Sentence Structure; Structural Analysis; Syntax

ABSTRACT

The report documents results of a two-year R&D effort directed at the completion of a prototype system for Chinese-English machine translation of S&T literature. The system, designated QUINCE, accepts Chinese input exactly as printed, with no pre-editing of any kind, and produces English output on experimental basis. Coding of Chinese text via Chinese Teleprinter System (Model 600D) and subsequent conversion to Standard Telegraphic Code for a direct input to the computer have been successfully completed. Manual keypunching and coding have been completely eliminated. Error rate from coding raw Chinese Texts into STC is less than .03 percent. Operation of QUNICE System is a six-step process, beginning with INPUT followed by analysis of Chinese (SEGMENTATION, LEXICON, GRAMMARS) and synthesis in English (TRANSFER, EXTRACT). Machine translation proceeds sentence by sentence as a global context. Five grammars are applied in a specific sequence. The system of multiple grammars, replacing a single monolithic grammar, has been in successful operation since May 1974. The lexical data base now contains a total of 82,330 entries in CHIDIC. A dictionary of 35,000 lexical items on nuclear physics (PHYDIC) has been machine-selected from CHIDIC. The system's design is stable, immutable, and highly machine independent. Programming strategy emphasizes the requirements of flexibility, reliability, compactness, efficiency, and portability. (Author/KM)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED 108 513

RADC-TR-75-109
Final Technical Report
April 1975

CHINESE-ENGLISH MACHINE TRANSLATION SYSTEM

University of California at Berkeley

Approved for public release;
distribution unlimited.

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, New York 13441

FL006984

U.S. DEPARTMENT OF HEALTH
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

This report has been reviewed and approved for publication.

APPROVED:

ZBIGNIEW L. PANKOWICZ
Project Engineer

APPROVED:

HOWARD DAVIS
Technical Director
Intel & Recon Division

FOR THE COMMANDER: John P Huss

JOHN P. HUSS
Acting Chief, Plans Office

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-75-109	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* CHINESE-ENGLISH MACHINE TRANSLATION SYSTEM		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 1 Sep 72 - 31 Aug 74
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) Dr. William S-Y Wang Mr. Stephen W. Chan, et al		8. CONTRACT OR GRANT NUMBER(s) F30602-73-C-0119
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California at Berkeley 118 California Hall Berkeley CA 94720		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS IDHS0401
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRDT) Griffiss AFB NY 13441		12. REPORT DATE April 1975
		13. NUMBER OF PAGES 204
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* Same		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer:
Zbigniew L. Pankowicz (IRDT)
AC 315 330-4231

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Computation Linguistics
Chinese-English Machine Translation
Structural/Descriptive Linguistics
Automated Lexicography
Computer Programming

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The report documents results of a 2 year R&D effort directed at the completion of a prototype system for Chinese-English machine translation of S&T literature. The system, designated QUINCE, accepts Chinese input exactly as printed, with no pre-editing of any kind, and produces English output on experimental basis. Coding of Chinese text via Chinese Teleprinter System (Model 600D) and subsequent conversion to Standard Telegraphic Code for a direct input to the computer have been successfully completed. Manual keypunching and coding have been completely eliminated. Error rate from coding raw Chinese Texts intc STC

is less than .03%. Operation of QUINCE System is a six-step process, beginning with INPUT followed by analysis of Chinese (SEGMENTATION, LEXICON, GRAMMARS) and synthesis in English (TRANSFER, EXTRACT). Machine translation proceeds sentence by sentence with a limited amount of information retained from sentence to sentence as a global context. The System consists of 5 components (total of 27 modules). Linguistic data base contains 3,460 programmed rules, of which 2,760 control the analysis of Chinese and 700 the interlingual transfer from Chinese to English. The battery of grammars, incorporating analytical and interlingual transfer rules, includes 5 grammars applied in a specific sequence. The system of multiple grammars, replacing a single monolithic grammar, has been in successful operation since May 1974. Implementation of multiple grammars greatly improved the capability of QUINCE System to correctly parse a great variety of Chinese sentences, eliminating or reducing -- at the same time -- multiple syntactic interpretations for each sentence. Lexical data base now contains a total of 82,330 entries in CHIDIC (30% in general language vocabulary, 60% in physics vocabulary, and 10% in vocabulary of chemistry and popular literature). All English glosses in CHIDIC have been extended to allow retrieval of interlingual information and editing during the EXTRACT step of machine translation processes in the computer prior to printing out of English output. Semantic feature notation, introduced into lexicography, significantly enhances the System's capability for in-context resolution of ambiguities. Updating of the general language vocabulary in CHIDIC is now at a minimum. One of the last update batches of 2,000 new accessions consisted of 100 general language entries (5%) and 1,900 S&T terms (95%). The System's design is stable, immutable and highly machine independent. Programming strategy emphasizes the requirements of flexibility, reliability, compactness, efficiency and portability. The System is invulnerable to a transition from one computer enviroment to another, and can be easily adapted to any third generation, large scale computer with requisite I/O media.

TABLE OF CONTENTS

Page

Abstract 1

I. Introduction 3

1. Programming 4
2. Subgrammars 7
3. Lexical Compounds and Productive Compounds 11
4. Chinese Character I/O 13

References 18

II. Programming System Development 19

1. Overview of the Berkeley Machine Translation System 20
2. History of System Implementation 25
3. Quince System Implementation Goals and Techniques 29
4. Quince System Programming Language and Translator 36
5. Quince System Data Definition Language Compilers 38
6. Quince System Report Program Generator 43
7. Quince System Storage Definition and Initialization 44
8. Quince System Optimization and Monitoring Techniques 46
9. Evaluation of Quince System Implementation 51

References 55

III. Lexicography and Text Organization 62

1. Accession of New Dictionary Entries 62
2. Random-Access Dictionary 65
3. Concentric Dictionary Organization 66
4. Dictionary Updating and Maintenance 67

Page

5. Dictionary-Based Research of Chinese Morphology 72

6. Chinese Input and Output 80

7. Chinese Character Data Base 83

8. Telecode Substitution and Text Normalization 91

References 95

IV. Studies in Interlingual Syntax and Semantics 96

1. Semantic Research 96

2. Linguistic Analysis vis a vis Interlingual Transfer 113

3. Higher Syntactic Structures: Noun Phrases 174

V. Conclusion 185

Appendix 1: Computer-Plotted Displays of Levels of Chinese Sentence Structure 190

Appendix 2: Style Differences in Modern Written Scientific Text 201

TECHNICAL EVALUATION

"Chinese-English Machine Translation System"

The report contains a detailed technical documentation of a two-year R&D effort resulting in completion of a prototype version of QUINCE System for machine translation of S&T literature from Chinese to English. The entire effort consisted of three major tasks described in Section II (PROGRAMMING SYSTEM DEVELOPMENT), Section III (LEXICOGRAPHY AND TEXT ORGANIZATION), and Section IV (STUDIES OF INTERLINGUAL SYNTAX AND SEMANTICS).

Software design philosophy and programming strategy deserve a special recognition as an important accomplishment in the field of machine translation R&D. The System is now virtually invulnerable to a transition from one computer environment to another in the realm of large scale, third generation computers. Adaptation to any particular computer environment requires only minor and relatively trivial adjustments because the System is highly machine independent with respect to the programming language and computer characteristics. The System is furthermore open ended, and provides facilities for both R&D and production.

Linguistics and lexicography of the System are based on a judicious combination of theoretical and pragmatic approaches to machine translation. It is significant, however, that the most satisfactory features of the System owe their existence and permanency to a sound foundation in linguistic theory. Nevertheless, if conclusive theoretical solutions are not immediately available, the System employs heuristic procedures reflecting the best current knowledge of Chinese and English. A considerable proportion of such procedures will probably acquire a permanent status on continuing exposure to a great variety of Chinese S&T tests. Other such procedures are workable, but of interim nature, and will be eventually replaced by generalized theoretical solutions. This R&D strategy allows an ever-increasing reliance on permanent, rather than *ad hoc*, solutions, and offers the following advantages. Optimization of the System in operational environment will neither persistently confront the user with

ACKNOWLEDGEMENT

The authors were Dr. William S-Y Wang and Mr. Stephen W. Chan; the contributors were Mr. Stephen P. Baron, Mr. Jerry Derkman, Mr. Harold Clumeck, Mr. Herbert Doughty, Mr. Cheung Fung, Mr. Robert Gaskins, Mr. John Hou, Mr. Robert Krones, Mr. Philip Robyn, and Miss Stella Ting.

numerous basic R&D problems still unresolved, nor will it ever require a virtually endless addition and/or revision of ad hoc solutions in a futile attempt to compensate for deficiencies in linguistic theory. There is no conflict between R&D and operational applications. In fact, both activities are interdependent and mutually complementary. Operational readiness will not depend ad infinitum on a theoretical solution of all linguistic problems encountered in S&T literature. This R&D strategy allows, therefore, an orderly and systematic refinement of machine translation quality, resulting in a steady and predictable enhancement of cost/effectiveness ratio in production of machine translation.

ZBIGNIEW L. PANKOWICZ
Technical Evaluation

ABSTRACT

This report documents further progress and results of a two-year effort in developing the Chinese-English Machine Translation System. The major design principles were reported in a previous Final Technical Report (RADC-TR-74-22, Feb. 1974), under contract No. F30602-73-C-0119. During this period, the major modules of the system have been programmed and separately tested. The integration and check-out of the complete system will still have to be carried out.

Several innovative procedures in system programming for large complex systems have been incorporated. Standards for controlling the quality of coding were set up and checked by a system language GASP which enables the programmer to write in a high-level language and produce the output as a Fortran program. Great care was taken in the design and implementation to ensure maximal portability of the system from one installation to another. Those segments of the system which must be installation-specific have been clearly identified to enable efficient conversion. Every care was taken in coding the programs so that each module can be easily interfaced. The principles of structured programming have in general governed the design and implementation.

The number of lexical entries in CHIDIC now totals 82,330, with 60% in physics vocabulary, 30% general vocabulary and 10% from chemistry and

popular literature. Coding of Chinese texts via the Chinese Teleprinter Model 600D and the subsequent conversion of this code into standard telegraphic code was successfully completed. Manual coding and keypunching has been completely eliminated. The error rate from the coding of raw text to telecodes ready for input is less than .03%

In the area of language analysis and rule implementation, there was also a significant advance in the Project's ability to implement a series of subgrammars, whose total effects were not only equivalent to the previous large monolithic grammar but also showed improvements in ability to parse a greater number of Chinese sentences accurately and at the same time decreasing or eliminating ambiguous analyses for each sentence.

Gradual shifting over of CHIDIC entry coding into a feature notation received detailed study. In order to be able to have a useful version as early as possible, the decision was made to concentrate this study on the lexical items selected from a corpus of 430,000 characters of nuclear physics texts, resulting in the creation of a smaller dictionary named PHYDIC which currently contains 35,000 entries. While the CHIDIC file is still accessible to the system, the emphasis has been on depth rather than breadth in the accession of entries in PHYDIC in order to reflect as soon as possible research results on syntax and lexicography, which lead directly to better output quality.

I. INTRODUCTION

This chapter will give a general overview of the work on Chinese-English machine translation research at the University of California at Berkeley during the 1972-1974 contract period. The research performed in this period continued the further development of the MT system which was presented in our previous Technical Report (RADC-TR-74-22). The major areas of development were: (1) programming of a new system using the currently accepted techniques of structured programming. The result was not only an increase in the efficiency of program writing and maintenance but above all a most desired capability of the system for being easily adaptable to the needs of linguistic analysis. (2) The restructuring of the grammar into smaller subcomponents. It parallels the programming effort by enabling the linguist to maintain the system of rules in a humanly possible fashion. This provided the capability of writing sets of rules in the form of 'subgrammars' to account for the detailed analysis of particular areas of syntax. It enabled the linguists during this period to clearly delineate differential effects on syntax between classical and modern written Chinese which can exist within a single sample of scientific writing. (3) Dictionary organization was revised by planning for a series of smaller specialized dictionaries which could quickly and easily incorporate information resulting from linguistic research. Instead of using the large CHIDIC file actively for lookup, we have selected lexical items from it to form a new nuclear physics dictionary containing 35,000 entries. This new dictionary (PHYDIC) was machine-selected using a corpus of 430,000 characters of machine readable Chinese nuclear physics text. Linguistic development will primarily concentrate on PHYDIC. Both CHIDIC and PHYDIC will be kept strictly compatible so that CHIDIC will always serve as a reserve.

Each of these developments are discussed in detail by a member of our Project: The programming system by Robert Gaskins; the lexicography and the normalization of Chinese texts for machine processing by Philip Robyn; and the syntactic and semantic interlingual analysis by Stephen Baron.

The present chapter will serve as a guide to the detailed material in the remaining chapters of this report.

1. Programming

The Syntax Analysis System (SAS) which successfully preceded the current new design of the QUINCE system continues to provide a framework with which linguistic research could continue uninterrupted while QUINCE is undergoing program development. By adding as extensions to SAS those components which are planned to go into the QUINCE system, it has been possible for us to tap the resources of the SAS in furthering our linguistic studies. It is true that certain inconveniences during the transitional stage had to be tolerated, but the advantage has been that both programming and linguistic research could be carried on in parallel to provide a generally smooth transition during this stage. Thus was evolved the concept of the SAS-QUINCE dual system. As soon as a new component in QUINCE was readied and, if it was not extremely difficult to tag onto the SAS, this grafting was done to test out the new capabilities of the new component or at least their intended effects. For example, the 'four-way lookup' package for QUINCE, which will greatly improve lexical look-up was slightly modified and replaced the single direction lookup scheme existing in SAS. The performance of subgrammars, which can only be done completely in the QUINCE system, was at least partially imitated by SAS. In this case it was necessary to make several intermediate runs on SAS in order to achieve the result which QUINCE can provide in one smooth execution. However, the full set of data structures for QUINCE was not yet available. The runs using SAS, though highly inconvenient, provided important and early feedback to the linguists, giving tangible proof that the QUINCE implementation will perform as designed. Appendix 1 is a detailed description of the application of subgrammars using the SAS. Plotting capability exists at present only on the SAS and this has proved of tremendous aid in the design of subgrammars.

Program design and implementation always kept in view the prime desiderata of flexibility, reliability, compactness, efficiency and portability. These are discussed in detail under QUINCE System Implementation Goals and Techniques (Chapter 2.3). For even though the main thrust of our effort is concentrated on implementation on the CDC 6000 and IBM 360 series on the Berkeley Campus, there are even constant changes in system architecture at these installations which would affect systems of programs as complex as ours. For example, although all the essential components of our system had already been coded as separate modules, the Project had to revise a major portion of its front end system dealing with accessing of dictionary entries during look-up. This came as a result of the decision to abandon our original favored design of a random access dictionary residing on disk and to revert to a sequential access system on tape. This constraint was imposed on us due to the unavailability of disk hardware at the Berkeley Computer Center. Although we have always been told that disk storage would be up within a short foreseeable period, it seemed wiser not to depend on merely promised hardware for the further development of our system.

It is a reflection of the robustness and flexibility of our general system design that a switch to a different mode of access does not substantially affect any of the other components of the system. It means of course that a great deal of program rewriting was necessary (and still going on at this writing), but the task is clearly delimited. If and when disk storage is available, the system can again make use of random access files without interrupting the function of the system as a whole.

Since the QUINCE system is developed with eventual production capabilities in view, aids to optimization in the current version have been built in so as to monitor as closely as possible portions of programs which can later be slated for optimization. Among others, this includes the counting of loop executions as well as the tallying of real elapsed time for specific routines. The importance of this type of monitoring in a developing system cannot but be emphasized. This will later serve as a highly useful aid for the estimation of throughput in a production system.

2. Subgrammars

As work on our grammar progresses further and further into ever more detailed formulations of syntactic rules, the burden of the linguist in keeping track of all the intricate rule connections becomes ever more severe. This was especially so because the form of the grammar developed so far consisted of context-free rules all gathered in one monolithic structure. Application of the rules is essentially simultaneous in the sense that all rules which could apply will be applied. Since it is well-known in the linguistic literature that there are different levels of complexity in syntactic structure, one is not surprised that certain ambiguous or even unwanted structures would have emerged under these conditions of analysis by the system.

It was the awareness of this hierarchical structure in syntax and the desire to reflect this in the grammar without embroiling ourselves in the manifold problems of rule ordering that led us to restructure our single large grammar into a series of smaller subgrammars. Each subgrammar will become a 'specialist' (to borrow a term from Winograd (1972) dealing with specific aspects of syntax. For example, there can be a set of rules for dealing with the classical stratum of morphology; another set of rules may deal with certain directional complements of verbs; or there will be rules dealing with the formation of the Chinese numeral system, etc. Each set of rules is applied separately and is seen by the Parser as a complete grammar. Some of these subgrammars may interact and some are independent. Currently the various ways in which this system of subgrammars will interact still need further specification and research. However, the most tangible results have not been in the theoretical implications but in the practical aspects of our research. Linguists have found it much easier to keep track of or concentrate on writing rules for a particular subgrammar without the constant worry of getting entangled in other rules; we are now able to draw more specific boundaries regarding the effects of certain sets of rules and to see more clearly where other rules must necessarily interact. Although it is true that with regard to bookkeeping, several thousand rules is no problem to keep track of for a machine, it is after all the human linguist who must constantly work with these rules. A methodology which makes the human linguistic analysis more efficient should certainly be exploited to its fullest extent.

But there is another respect in which the separation of these rules into subgrammars will pay dividends in computational facilitation. The literature in Artificial Intelligence research (cf. Nilsson, 1971) stresses the necessity for intelligent search techniques, especially with respect to heuristic techniques for arriving at solutions which will be attained with the least number of computation steps. In a machine translation system, where a great deal of linguistic, contextual and pragmatic information are still not available within the system, the implementation of many heuristic and semi-heuristic processes is important in the resolution of those ambiguous analyses generated by the system in its initial stages of analysis. In the earlier version of our system, the tightly packed set of grammar rules allowed little flexibility in adding heuristics within its framework. However, by having clearly delimited sets of subgrammars, one may 'insert' heuristics at the interface of one subgrammar and the next. Thus it will be possible, for example, to take stock of the partial tree tops produced at the end of one application of one subgrammar, by setting up priorities with respect to the types most likely to occur at that stage and trimming the other nodes which have lower priorities. A number of evaluation techniques may also be made use of at this stage to aid in making heuristic decisions. For example, since lexical items could have certain priorities attached to them (such as differentiating a multiple-function item in terms of its syntactic and semantic functions or its use in a particular discipline; frequency of occurrence in colloquial versus literary usage; classical versus modern written style, etc.). These priorities may then be evaluated over the whole subtree and compared with evaluations over another subtree. Thus a node with a lower priority would be discarded, eliminating some of the alternative structures which would have otherwise cluttered up the analysis.

Besides this backward glance by the heuristic mechanism at the end of a subgrammar application, similar mechanisms can be constructed to look forward to the application of the next subgrammar. After the previous node trimming procedures have been carried out, the nodes which will take part in the application of the next grammar would still be carrying their priority measures. It may be more effective to specify particular rules which have to be applied for a certain lexical item or a particular node.

(In our scheme, terminals and non-terminals can be viewed in the same way, since the non-terminals, especially the top nodes resulting from the application of one subgrammar, will essentially be treated as terminals for the next subgrammar.) This call for a particular rule can in turn trigger the call for a particular subgrammar. This lookback and lookforward mechanism can become quite powerful, enabling many decisions to be made in the course of parsing. The attractiveness of this approach is that the application of a set of rules and the heuristics have well-defined boundaries. The subgrammars are clearly formulated rules of the language, couched in a phrase structure format; whereas the heuristics can assume freer forms that are more flexible in terms of implementation, such as having them as separate procedures dealing with particular aspects of the language. However, since the interaction between heuristics and the subgrammars is kept clean, there will be an overall advantage in ease of revision, both to the grammar and to the heuristics of the programs themselves.

In Appendix 1, the results of one sentence which was analysed by the current set of subgrammars is presented. The results were produced by our SAS version of the system since the full set of programs for displaying intermediate results was not yet ready for the QUINCE system. There are minor adjustments in SAS but the essential parsing processes of the QUINCE system are logically presented, giving us a clear picture of the implementation of subgrammars. It should be noted that only subgrammar rules were applied. No heuristics have been implemented at this stage of development.

3. Lexical Compounds and Productive Compounds

The problem of compounding is a highly complex issue for any language. It ranges from consideration of rule-oriented morphological processes to semantic, contextual processes of which linguistics has not yet even obtained a firm grasp. In dealing with Chinese compounding the problems are presumably more severe because the paucity of explicit morphological cues has thrown the burden of analysis onto the shoulders of the semantic component. As the latter is still in an exploratory stage, few algorithmic statements can be made; and this naturally makes machine processing in this area even more difficult. However, in our own efforts at tackling this problem, we have been able to make more explicit certain processes of compounding in Chinese. Some processes which were considered productive have now been identified as essentially lexicalized, especially with respect to bound monosyllabic nouns belonging to a classical stratum. Other modifier-head compounding processes have been studied in detail and we have discovered certain cross-over restrictions in modification which will in fact help to reduce the number of possible modifier-head compounding possibilities for a particular noun-noun pair. This has a most salutary effect on the total capability of our system in arriving at correct analysis, since we have found a majority of so-called ambiguous analyses were the results of a lack of detailed knowledge of the syntacto- semantic properties in Chinese. Extensive discussions of this aspect of our research will be found in Chapter 4.

It has also been possible for us to revise by machine the entries of long compounds in our dictionary to conform with these new findings.

Essentially the longer entries were treated as text strings and submitted to our SAS for analysis using the grammar and a modified dictionary which does not contain the entries in question. This is like a bootstrapping process whereby all entries in the dictionary which are capable of being analyzed by the rules of the grammar are eliminated from the final form of the dictionary. It means that all productive processes will be handled by grammar rules, and only truly lexicalized entries will be allowed in the dictionary. This method of large-scale machine revision is further discussed in Chapter 3.

Finally, since one cannot deal only with simplex sentences in text processing, considerable effort was spent in formulating rules to account for complex sentences. As has been mentioned often in our earlier reports, the printed text in Chinese, because of its punctuation characteristics, does not give clear indications of what might constitute a sentence, especially a complex sentence. A great deal more research into the properties of these "higher syntactic structures" is needed, especially the ability to encode some type of intersentence information. For example in order to render the correct tense in English, it will be necessary to determine the aspectual reference in the Chinese text. This may not be directly obtainable at the sentence level. The Project has, however, studied the time relations expressed by sentential adverbs and has classified these by types, such as adverbs expressing duration versus point of time, calendar time, repetitive time, etc. Some adverbs would only occur in a presubject position while others can occur in pre- and post- subject positions. This type of information is very useful in cases where a certain noun phrase is set off by commas on both sides and it is necessary to determine its function in the context of other syntactic units. It is a frequently occurring phenomenon in Chinese that the object is preposed to the beginning of the sentence and, in the absence of other semantic criteria, may well be taken to be the subject.

Further work in intersentential relations was also carried out in terms of logical precedence in time and action between two adjacent sentences (Chan,1975), or between subordinate and coordinate structures. These are still in the exploratory stages. It seems that the emerging interest in more vigorous descriptions of "text grammars" such as the works of M. Gopnik (1972) and Van Dijk (1972) will be helpful.

4. Chinese Character I/O

The coding of Chinese characters into Standard Chinese Telegraphic Code ('telecode') has been the basis of machine processing of Chinese material since the inception of Chinese-English MT. However, since machine processing requires that everything be coded in linear, sequential form, there are quite a number of problems to be solved in order to capture all the information presented on a printed page. For example, non-Chinese symbols occur extensively either as alphabetic symbols or numerical symbols. Mathematical and chemical formulas need special formats which in fact have their own 'grammar'. In order to have a consistent telecode form of representation, 4-digit or 4 alphanumeric characters must be used to code any special symbol. Since no international standards are in use in all cases, the Project has assumed the task of standardization for internal consistency. (It is also crucial that the telecode form of representation be used throughout since the system will only accept this type of machine coding for text.) However, this task amounts to, in fact, reducing any type of printer's symbol (such as any alphabet in any language) to a 'telecode-like' representation. Theoretically, this is a very discouraging prospect and eventually a more direct graphical method of representation will be the way out of this dilemma. This of course involves future efforts in optical character recognition - an area in which Chinese text processing is just crying out for assistance.

Be that as it may, the Project has been able to code texts in a more advanced state than hand-coding by telecode. Our previous final report discussed in detail the use of the Model 600D Chinese Teleprinter System for mechanical coding of texts. Further refinements to this system of coding have been made. We were able to write software which will convert the Model 600D code to system acceptable telecode. A half million characters were converted in this way. The error rate was a mere 0.03%, which for all practical purposes is 100% accuracy starting from the human keyboard operator to the end product of system compatible telecode.

There is an understandable tendency for the prospective user of the Chinese-English MT product to minimize the importance of Chinese character ouput as an integral part of the whole MT process. From the user's point of view, he being someone who has no knowledge of Chinese (or he would not need the translation), the whole question of character output is irrelevant. But it is not clear whether this requirement is truly unwanted or that users simply have not had any experience in using translated Chinese material produced by machine. For example, when a Russian or German word had not been looked up or was not available in the translation output, it is quite reasonable to print out the original or transliterated form of the word so that the user may have a chance to look up the word in a dictionary himself. The user has a one-for-one transliteration so that he knows exactly which word it is that has not been translated. One might assume that a similar transliteration may be possible for Chinese. This is partly true but at a great cost of effort. The only one-for-one 'transliteration' will be via telecode. But a telecode book is generally not a very handy reference work. And even when the telecode is available, ordinary Chinese-English dictionaries do not provide this information. The Chinese characters still have to be recognized first. 'Romanization', which is analogous to transliteration, actually only gives the phonetic representation of the Chinese character. The well-known ambiguity problems simply rule this out as accurate representation in the output. For scientific terminology, when very often low frequency characters are employed, there can be no substitute for knowing the exact original Chinese representation. Thus for practical considerations, the machine translation output device should be able to print Chinese characters as well as English. We are thus led to considerations of a very different type of output than that which current highspeed impact printers can normally provide. In fact, the most suitable device currently available will be the class of electrostatic plotters and similar equipment as well as CRT's. They provide the flexibility of intermixing alphabetic symbols with other graphic symbols of which Chinese characters will be our most interesting category.

In our current work, the need for character output is even more keenly felt since telecode is simply not an efficient means of communication by humans, aside from the fact that research is being performed by the linguist and lexicographer (who already have excellent command of Chinese orthography) at every stage of feedback--dictionary entry checking, text proofing, concordance searching, analyzed text sentences, etc.--all must be done via telecode. As the progress of research accelerates, this delay in the feedback loop is seen to be less and less efficient and less cost-effective. It is thus imperative that parallel research effort be continued to obtain character output for all the day-to-day work of the Project.

The development of a character plotting package which will be fully compatible with the design of our new system was planned and subcontracted to Dynamic Graphics, Inc. of Berkeley, a firm with whom the Project had previous good experience. The development of this package was, however, discontinued owing to the fact that Dynamic Graphics was unable to deliver the product on time in spite of a six-month extension, as well as a sudden change in further funding prospects. We were able, however, to obtain without charge a substantial amount of good code that has been written, and further development can continue in the future using this package. More detailed discussions of our research are contained in Chapter 3 The advantage of the current design for this package is that it uses off-the-shelf hardware and a special keyboard (the 600D Teleprinter) which is already in use at the Project. We are confident that further development of the character package will greatly multiply the productivity and the pleasure of research and development efforts in Chinese-English MT.

References

1. Chan, Stephen W. 1975. "Asymmetry in Temporal and Sequential Clauses in Chinese," Journal of Chinese Linguistics 2.3.

2. Dijk, T.A. van 1972. Some Aspects of Text Grammars. Mouton. The Hague.

3. Gopnik, M. 1972. Linguistic Structures in Scientific Texts. Mouton.

4. Nilsson, N. J. 1971. Problem Solving Methods in Artificial Intelligence, McGraw-hill, New York.

5. Wang, W. S-Y., S.W. Chan et al. 1974. Development of Chinese-English Machine Translation System, Technical Report RADC-TR-74-22, (AD776813). Rome Air Development Center, New York.

6. Winograd, T. 1972. Understanding Natural Language. Academic Press, New York.

II. PROGRAMMING SYSTEM DEVELOPMENT

During the period of this contract, programming development has centered on two distinct but related tasks: (1) devising and refining algorithms and processes for use in carrying out the steps of natural language translation; (2) embodying those processes in a new system of programs designed to serve as a robust base for further research and development efforts.

The general strategy for the Berkeley system was designed and ready for elaboration at the beginning of the contract period. Detailed development since then has not revealed any reason to change that strategy, but in the process of refining each step we have succeeded in relating our conceptions to generally-known formalisms of parsing and translation, which has in turn suggested fruitful areas for later extensions. By and large, however, implementation of these extensions has been deferred until the basic system is complete in a new system of programs.

The actual construction of the new programs, on the other hand, has been undertaken in a framework devised largely during the present contract period. The problem of how to code programs as large as those for machine translation in a way which will make them flexible, portable, and maximally transparent has led us to adopt a number of innovative programming methods which are of independent interest, and which have already proved to be of exceptional worth in increasing the productivity and manageability of our programming. Much of our confidence in our ability to make use of ever-increasing linguistic knowledge in the future to improve the capabilities of actual running programs can be attributed to these innovations in programming methods.

1. Overview of the Berkeley Machine Translation System

As a framework for later discussion, we give here a general overview of the principal sub-processes of the Berkeley system. More detailed discussion of the design issues can be found in previous technical reports of the Project (Wang & Chan, 1974), and the linguistic motivation is the subject of other sections of the present report.

One fundamental principle underlying the Berkeley system is that there should be a clear distinction between the programs for translation and the linguistic data which they make use of. Thus, the programs themselves do not incorporate any specific knowledge of Chinese or of English; all of that is contained in various data bases for the programs to use. (This is not to say that the system is independent of the particular language pair Chinese-English, because the particular processes which we have chosen to implement first are those which we judge to be most effective for that specific language pair. It is true, however, that our general framework could be elaborated in different ways to work with data bases for other language pairs.) So with each principal sub-process we will mention briefly the kind of data bases with which it works.

Conceptually, from a computer scientist's point of view, the Berkeley system is organized by analogy to a syntax-directed compiler [6,12,43].

One cannot press this analogy too far, because natural languages seem to exhibit none of the simple properties of programming languages which make their syntax-directed compilation so easy. But early machine-translation systems were arranged like the earliest compilers, as a series of ad-hoc recognition and translation steps; the Berkeley system is organized in the manner of a modern compiler: (1) it accepts a string in the source language; (2) it divides the string into atomic lexical tokens, and (3) it assigns a syntactic structure to them, using a grammar, to produce a syntax tree; (4) the syntax tree is converted to a structural tree for the target language, using semantic information associated with productions of the grammar together with a limited amount of global information, and (5) a resultant string in the target language is extracted from the transformed tree. In a translator for a programming language, each of the five steps is straightforward; but in the Berkeley machine translation system, each step has its own additional complexities, as we shall now see. (Each of the five steps will be identified in this outline by its general name in the Berkeley system.)

Step 1. SEGMENTS. The task of identifying a suitable string from the Chinese for processing is accomplished by consulting a table of punctuation and fixed words, to break the text into "sentences". (Very little is yet known about "discourse grammar", or the organization of texts at a level higher than the sentence; some beginnings in this direction in a machine-translation context are given in a recent report by our colleagues at Berkeley's Contrastive Semantics Project (cf. Wallace L. Chafe, "An Approach to Verbalization and Translation by Machine" [RADC-TR-74-271, October 1974]). For Chinese and English, however, it is necessary to proceed further on a variety of heuristic cues, to divide the sentence into a number of smaller sub-sentences which will be the primary units for parsing; the sub-sentence segmentation must be done tentatively, subject to later reversing.

Another complication is that the input "source string" cannot really be considered a string, since some of its symbols (each representing a Chinese character) are "cover terms" for any of several alternative characters. Hence all these alternatives (derived from a one-to-many mapping in "substitution tables") must be preserved for later lexical verification.

The effect of this is that even the "input string" is not a linearly-ordered string, but rather a partially-ordered set of symbols; by the customary slight abuse of the term, we refer to this as a "lattice" of symbols. So the result of SEGMENTS is a list of lattices of symbols for a sentence.

Step 2. LEXICON. The next step is to identify lexical tokens in the input and to assign them to lexical categories. This is made difficult by the fact that written Chinese contains no word-delimiting symbols; hence, the general solution is to look up all the substrings of the source lattices in a Chinese dictionary, to see which are possible lexical sequences. The division into lexical words accomplished in this way is not unique (in general), so alternative partitionings into lexical sequences must be preserved for grammatical resolution.

It is also true that an astonishing amount of apparent lexical ambiguity exists for Chinese (alternative syntactic categories for a single possible lexical terminal, each with its own set of grammatical and semantic properties) and these too require syntactic treatment; so the best that LEXICON can provide is yet another list of lattices, lattices of possible lexical tokens, for parsing.

Step 3. GRAMMARS. Given these complexities, the way in which parsing must be elaborated is tolerably clear: the parser must parse (in parallel) all the possible paths through terminal lattices, assigning a grammatical structure to the admissible sequences and rejecting inadmissible sequences, thus resolving some of the alternatives left open by earlier steps.

Additional requirements enter here, however. First, the sub-sentence parts produced by SEGMENTS do not (and should not, often) parse to a single non-terminal; instead, the result of parsing will be a lattice, each of whose elements is a partial parse tree. Only when the segment-parses are combined, can still another grammar establish admissible sequences of partial parse trees.

Second, this situation coincides with a linguistic desire to break (what would otherwise be) a very large grammar for Chinese into specialized sub-grammars which can be applied one after another to a segment or a combination of segments. So, the Berkeley parser actually works by parsing a lattice of trees into a lattice of trees; any parse result is itself a suitable object for parsing, alone or concatenated with other lattices.

Although this general line of thought is clear enough, the detailed working out of the multiple-grammar parsing algorithms in a way which would interact with translation cleanly and which would admit of efficient and modular implementation is one of the principal results during the current contract period.

Step 4. TRANSFER. Having completed the analysis of the syntax of the Chinese input segment, the next step is to begin the synthesis of the English sentence (or sentences) which will be its translation. The process is closely akin to what computer scientists understand as a "syntax-directed translation" in formal language theory [1,24]: a new tree is computed as a function of the old tree and its derivation history (here, its "recognition history"). The Chinese lexical terminals have translations as single English lexical items, or as (sibling) sequences of such terminals, or as nothing. The translation of non-terminals is governed by the grammatical production expanding them, and may produce a corresponding sub-tree with a permutation of descendants, deletions, insertion of constant trees, and copies of other subtrees as descendants.

(This is a fairly "classical" set of restrictions on the translation relation, but it answers to current linguistic needs fairly well. There are a variety of more complex translation schemes which could be of potential use, and exploration of how they may be linguistically exploited is a task for the future. There are a number of problems connected with computing the translation of an ambiguous tree, since the translations may differ in unpredictable ways from one another: for this reason, a translation scheme of middling complexity may often be impossible to carry out. A potential line of development would be to impose well-formedness conditions on the range of the translation relation, and thus use it to prune in the domain. Our thinking is along the line of references [18,28,29,39,46].)

Step 5. EXTRACT. The final step is to use the tree resulting from the translation to produce a string of English words, as the result of the whole process. This step of "string extraction" must collapse strings which stem from different syntactic analyses but which are identical, while preserving true ambiguities of the language, so the process is more complex than the usual one of simply reading the terminal frontier of the translation tree. It would be in this step, too, that morphological adjustment of the result text would take place; but since the Berkeley system does not yet have a data base for English with which to carry out that process, it has been deferred. (A possible source for such a data base will be the data from the University of Texas LRC-German-English machine translation work.)

This--in outline--is the design of the Berkeley system. Just by considering its relation to the skeleton of a compiler, it is clear that any implementation of these capabilities will result in a very large and very complex set of programs. How implementation results have been achieved during the contract period is the subject of the remainder of this section.

2. History of System Implementation

As of the beginning of the contract period, a "first approximation" to the new programming system existed, in the form of the code for the "Syntax Analysis System" (SAS). The code for the SAS has been begun in 1967, and it had been producing output since 1968. The SAS was written to accomplish initial aims quickly, without specifically attempting to anticipate every possible linguistic development for the future; nevertheless, the SAS proved able to sustain extensive additions and changes over the next several years, so that by the beginning of the contract period it had migrated to a state recognizably close to the planned design for the new system of programs.

While the SAS was highly encouraging, there were several features of the new system which went substantially beyond what was reasonable to add to the existing code (pre-eminently, multiple-grammar parsing). It was also clear that the extension capability of the SAS was being exhausted, so that the work involved in adding new functions or making changes was too great to allow really easy linguistic experimentation. Considerable attention has been paid in the last two or three years to this problem of large programs becoming less manageable, and it seems that much has been learned recently [5, 8, 9, 10]. In retrospect it appears that the SAS programs were probably much better than average in their capacity for growth, but they had not entirely escaped the perils of their era.

A significant deficiency of the SAS programs was that they had been developed to make use of special peculiarities of their local language processors, loader, file system, and operating system, and strongly assumed the CDC 6000 - series hardware of the Berkeley Campus Computer Center. It was clear that the programs would be of greater applicability if they could be made independent of specific hardware. But this actually presumed a rewriting of the programs from the ground up to assure machine independence.

All these factors together led to the creation of an entirely new system, actually a highly robust and adaptable family of systems, which could be configured for a research environment or for production purposes, to run on a variety of large-scale hardware, and which could evolve easily and rapidly in any of several directions. It is the creation of this system to replace the SAS, called the Quince System, which has been the most visible activity under the present contract.

It was appreciated from the first that the Quince system would require some time to prepare. In order that linguistic research and experimentation not come to a halt during this period, it was also decided that simultaneously, crucial features of the new system should be incorporated into the old SAS. The SAS modifications would be performed as quickly and easily as possible, even at the cost of finally using up the old system's extension capability, and accepting cumbersome operation if necessary-- while at the same time the corresponding parts of the new Quince system would be prepared as carefully and systematically as possible.

This dual-system approach has had a number of practical benefits. Most important, perhaps, has been that the initial versions of Quince system capabilities embedded in the SAS have been available rather quickly for linguistic experimentation, so that preliminary feedback from experience can be gained even before the first full-scale implementation, serving often to clarify the design goals and to suggest an order of priority for Quince system coding. We have also been able, in the cases where it was warranted, to use the SAS programs as test data generators for the Quince system modules, and also to use the SAS display capabilities to verify Quince system output.

The costs incurred in dual-system development--for there are costs as well as benefits--have been rather different than we anticipated. We were on guard against a large diversion of effort into code destined to live a short life, but that danger never materialized; it has not been difficult, once a Quince system function was clearly specified, to find ways to provide simulations of its important properties (there are one or two exceptions, which were simply not provided in a dual version). Instead, the largely-unanticipated investment has been in modifications to a wide array of auxiliary programs which perform analysis or maintenance of data bases. In most cases the Quince system formats for data bases were so much easier for humans to work with that it seemed unreasonable to prepare data in new formats which would be translated into the old formats (though that, too, was necessary)--rather the new formats should be used throughout. We had known, of course, that in any case our programs for concordances, indexes, listings, updating, and so forth would have to be converted eventually; but since the SAS code would be running for some time, it had seemed that conversion could be stretched over a long period. That turned out not to be the wisest way to proceed, and so we went ahead to provide a full conversion of a great many of our auxiliary programs, using more resources than had been intended during this contract.

The net effect, then, of proceeding in this way has been that a functioning system has been continuously available for linguistic research, throughout the contract period, while preparation of a new programming base to assume those functions in a way which provides a firm base for future development has proceeded deliberately. This kind of smooth transition is of such importance in a research and development environment, that we judge the dual-system to have succeeded in its aims during the contract period.

The next question is, what are the advantages to be gained at the end-point of this transition?

3. Quince System Implementation Goals and Techniques

There are a number of design objectives which must be harmonized in the development of any large set of programs, and the larger the system and its data, the more important it becomes to build the satisfaction of the objectives into the programs from the first, leaving none of them to be treated as afterthoughts. The list of objectives is not particular to our machine-translation systems, but each has a particular application.

Flexibility. Since our programs are to support research and development, we expect that certain changes to the system will arise as research progresses. Thus, every technique for enhancing flexibility (modularity [36, 37], layered design [17, 30], higher-level languages [9, 19], debugging tools [41], parameterization in source code and in run-time organization) should be exploited to make frequent changes as quickly and cheaply as possible.

Reliability. One should anticipate that eventually all those programs will be running at a number of installations far from their developers, where significant cost will be attached to errors and to their correction. And in programs which change as frequently as ours, it is particularly important to be able to bring new versions to reliable operation quickly. Here what is wanted are coding techniques which facilitate testing [35, 49], provision of good testing monitors [22], and automation of the production of multiple versions to reduce the possibility of random errors from manual intervention [45].

Compactness. In common with all serious programs in computational linguistics, the data bases and intermediate results of our system are very large; it is simply essential to be able to pack and encode tightly, even at some cost in efficiency, and this must somehow, even in research versions of the system, be harmonized with flexibility and portability. We have given a great deal of attention to automating the compact storage of data.

Efficiency. As research versions of our system move toward production status, it is essential to have designed a way to make them run at acceptable speeds. An important way to do this is to be able to monitor execution to identify critical sections of code [31], and to be able to optimize selectively; in part this is a kind of flexibility, in part it is in conflict with it.

Portability. To achieve cumulative results in programming, it is essential that systems be easily moved among machines and operating systems; it is simply unacceptable to be forever rewriting the same system components for slightly different environments. Apart from thoughts of the future, we have had at least four environments to consider in our initial implementation: (1) The Berkeley Campus CDC 6400, which offers good and stable facilities for program development; (2) the Lawrence Berkeley Laboratory (AEC) CDC 6600's, which offer extremely low rates for production runs but with a modified and rather unstable operating system; (3) the UC North Data Center IBM 360/65 which provides local check-out of IBM versions, but which is a data-processing installation not oriented to program development; (4) the IBM 360/65 at Wright-Patterson, which had been nominated as our immediate Air Force user installation. Obviously, we might as well adopt a more general attack on portability than these four particular machines minimally require, since with very little more thought we can make our programs generally available and provide at the same time for future hardware changes.

There is, of course, the further requirement that solutions to all these requirements be susceptible to implementation by a programming group of the sort we want to maintain, a comparatively small group of extremely interested people; any approach which suggests many-fold multiplication of staff is not attractive, since programs should be both less expensive and more reliable than programmers.

Most of the rest of this section is devoted to setting forth the system production techniques which we have used to meet the requirements above. We feel that we have successfully reduced a number of "academic" computer science ideas to extremely practical use in producing large systems, thus remarkably extending the productivity of programmers.

The cornerstone of the approach adopted is serious allegiance to the philosophy (most notably associated with the name of Edsger Dijkstra [13]), that obvious conceptual clarity at every level of system design is imperative: that "elegance" is not only an academic luxury, but an indispensible requirement for large systems. The trick, of course, is to be able to afford such indispensible luxuries.

Accordingly, we undertook to design for our system a machine-independent representation which encourages flexible programming practices, and which can itself be implemented in a portable fashion, consistent with compactness and efficiency. This amounts to defining an "abstract machine" which is well-suited to running machine-translation programs, together with a way of transforming actual hardware machines to present that standard face to everything above.

The usual way of realizing that goal is to specify a programming language, and to write compilers for the language. In principle that is what we have done, but in a very economical way.

The desirable features in a language which we would want are the same as in any "systems programming" language, but the most important from our point of view are:

(1) Control structures to permit "go-to free" programming, in the sense of Dijkstra, Wirth, Hoare [13, 31, 47];

(2) Data specification permitting the use of heterogeneous structured data, along the lines of "Hoare records," together with some kind of based storage and allocation [13];

(3) Some macro facility to permit "stepwise refinement" without intolerable procedure overhead, isolation of machine parameters as manifest constants, conditional compilation to produce systematically-variant versions of system code [10,45];

(4) Provision for initializing variables and areas, including convenenient facilities for character strings;

(5) Efficient access to machine part-words and multiple-words as variables in the language;

(6) Access to all machine facilities and operating system primitives, at least through convenient access to external procedures coded in assembly language with a standard calling sequence [23].

There is certainly nothing controversial or even innovative about these requirements (in at least rudimentary form they were all recognizable in Jovial [38] many years ago), but still there is no standard programming tool which is at all satisfactory in these respects. The widely-implemented programming languages (Fortran, Cobol) and the fitfully-available PL/I are unsuitable, inefficient, or both; the academic languages are not widely available and suffer from toy implementations with poor access to host facilities.

In this unsatisfactory situation, we saw no choice but to fashion new tools. It was not our aim to produce full-scale industrial compilers for our ideal systems language on all machines (indeed, that was what we wanted to avoid), but a practical solution was to extend existing standard tools.

Having in mind particularly the example of the Bell Laboratories Altran system[21], we adopted as the "object language" of our system a selected subset of ANSI Standard Fortran [2,3,4] together with some rules to ensure its machine-independence; this would be impractical as a coding language, but it is catered for by reasonably good compilers on all machines of the class suitable for running our system [32]. We then provide a translator to translate our systems language into this ANSI Fortran subset, incorporating into our translator the desired macro capability. Where facilities which cannot be reasonably provided in this way are required--pre-eminently packed or structured data--our translator provides the syntax of a Fortran procedure call, and we must provide also a translation into a machine-dependent assembly language of an external procedure to answer the call.

The specific tools which we have implemented and are currently using, then, are:

(1) A translator from our "structured programming" language into an ANSI Fortran subset;

(2) Data-definition compilers; while having much in common with another, each of these has some features specialized to a particular machine, and generates code for a specific assembler. We currently operate two (one for CDC 6000/7000 machines, one for IBM 360/370 machines).

(3) A specialized program-writer which writes programs for the

translator (1), using the storage mapping provided by (2), to produce system-listings of data formatted properly without the use of a Fortran format interpreter.

(4) A storage-definer which takes a logical description of variables and areas, their types and initializations, and produces macro-bodies which can be expanded in all the translated Fortran routines needing them, as well as in CDC or IBM assembler routines generated by the data-definition compilers; it also produces loader directives (as Fortran Block Data) to perform initialization.

(5) A lowest-level package of run-time routines for error-handling tracing, tallying, timing, calls to which can optionally be generated by the translators and program-writers. These are again machine-specific; our versions for CDC machines are quite elaborate, for IBM machines more minimal, reflecting the fact that we currently do program development on the first kind of machine primarily.

On this base, we then proceed to erect machine-translation programs. All machine-independent procedures and all machine parameters can be specified to these processors, which translate, compile, expand, and finally produce one of a number of versions of the machine-translation system for a particular machine, in its own Fortran and assembler languages.

Implementation of this base for any machine is not quite trivial; but it is a well-defined and circumscribed task, very substantially easier than the implementation of the machine-translation system, and entirely independent of the logic of machine translation. The principal advantage of such a base over an even simpler one is that it yields very efficient and compact running systems, while not compromising the flexibility and reliability of the logical definition of the programs we are trying to write.

4. Quince System Programming Language and Translator

We call our system language "Gasp" (and that is the usual reaction to a proposal to carry out structured programming in Fortran); but "the sooner we can forget that Fortran ever existed, the better," says Dijkstra, and that is the aim of introducing Gasp.

Gasp fundamentally provides control structures to relate Standard Fortran logical and arithmetic expressions--and thus completely replaces all Fortran statement types except for (some) declarations, assignment statements, and procedure calls. Its appearance is approximately that of Wirth's Pascal [25], or the Project SUE language [11], with a smaller character set and a few more parentheses.

The variety of structures provided is very rich, catering to personal preferences consistent with go-to free programming.

For conditionals we use IF--THEN--(ELSE) and UNLESS--THEN--(ELSE), and two varieties of Hoare's Case statement: CASE--OF labels (ELSE) for sparse sets of labels, and SWITCHON--INTO labels (ELSE) for dense sets which can be translated into a jump table for best efficiency.

For repetitions we use WHILE--DO and UNTIL--DO for a loop tested at the top; REPEAT--UNTIL and REPEAT--WHILE for loops tested at the bottom; REPEAT--UNCONDITIONALLY gives a loop which can be exited in the middle. There is also a FOR statement of the usual kind to provide an iteration variable in the loop.

All of these constructions can be nested freely to any depth; any construction can be labeled, and can be exited unconditionally (EXIT) or conditionally (EXITIF, EXITUNLESS). Unlabeled exits refer to immediately-enclosing constructs, labeled exits leave the block so labeled and all enclosed blocks. It is not possible to use the exit mechanism to defeat the program structuring goals or to enter uninitialized blocks.

There is a macro-facility (recursive) using a keyword, and a text-string constant facility separately; both can be handled properly by a character-scan and a keyword classification, requiring neither token identification nor syntactic analysis by the translator; the translator is thus simple and fast.

A number of special facilities are provided. Fields in structured data items can be identified in Knuth-format using square brackets, such as

```
LLINK[RLINK[I]]  := LLINK[INFO[RLINK[J]]]
```

requiring translation into calls on storage-mapping procedures. Standard linking to error-handling is automatically generated.

There are a variety of debugging aids which can optionally be generated into translated procedures, by setting a toggle in the Gasp translator (these come up for discussion below). Some forms are also under programmer control; the form ASSERT (logical expression)--based on Wirth's Algol W--compiles into a test and error call in debug mode translation, and remains a comment in non-debug-mode recording assumptions and presumptions which can easily be turned into code again should doubt re-emerge as to their satisfaction.

Basically, the correct way to view the Gasp translator is as a macro processor [10], expanding calls on the control structure macros as well as programmer-defined macros. Some concession has been made to standard feelings about what the language should look like which would hamper its implementation directly as a simple macro processor, but neither does it require any general pattern-matching capability, and its implementation as a special-purpose translator is straightforward.

Some attempt is made to provide optimization of special Gasp features (such as deciding how much run-time code to compile to verify FOR-statement correctness), but the chief method of contributing to efficiency is to put out extremely stylized Fortran susceptible to machine-dependent optimization by a Fortran compiler [12,27,33].

5. Quince System Data Definition Language Compilers

One of the more successful projects in this approach to system development has been providing access to a "logically-organized" memory under Fortran; we have been able to keep our machine-translation programs entirely independent of all knowledge of real memory organization, while providing efficient storage-mapping through automatically-generated assembler code. The flexibility which this provides is amazing: we can freely at any time re-organize storage, move data from one class of storage to another, expand and contract tables and fields, change accessing methods--all by updating a few card images and re-compiling the data definition. Similarly, we can run a full-scale test on a module on our CDC machine, then carry a tape of the same card images across the campus and check out the same module on the IBM machine using much smaller data areas (to fit a week-day partition) without any manual adjustment.

To use the data-definition facility, a programmer defines a structure in (typically) five to twenty-five cards. He nominates a storage area by name, and specifies whether it is fixed or "floating" (based). He then specifies how many "nodes" he wishes to put in the area (fixed, or a variable number) and how much space a node requires. (Any number of node types, of up to two different sizes, may share an allocated area and be intermixed on a space-available basis.) We call such a collection of nodes in a single area a "table" (influenced by Jovial, here)--it is also like a "record class" in the Hoare terminology. An individual node may occupy from 1 bit up to 4096 characters of space.

The programmer then specifies whether he will do his own allocation (if, for instance, he is treating his table as an array--common with bit arrays), or if he wishes automatic allocation to be performed; in this case he gives the names of procedures which are to be generated to give out and take back nodes with proper initialization. If the program at run-time obeys a stack discipline, he gets a stack allocation, and thus he can locate twin stacks growing dynamically at opposite ends of a fixed area, to use the same space at different times. If stack discipline is not observed in use, a free-list is kept in the returned nodes. Several tables may be "chained" so that an allocation or return operation on the master table simultaneously affects all its related chained tables, possibly on different classes of storage.

Hash-addressed storage is provided as an alternative allocation. If a programmer chooses this, he specifies numeric and character-string fields of any length to be the keys. The format for accessing hashed tables is exactly the same as for an allocated table, except that a programmer simply uses his own keys instead of the "pointer" keys returned by an allocation routine. Hash tables may be either direct, or "indirect" (with an auxiliary compressed data table).

Next the programmer specifies all the fields in a single node, giving for each its name and a bit-level specification of its position and size. A field may be specified to hold a signed integer of a certain magnitude, a bit or bit-string (byte), or a character or character string of fixed length. Not all the nodes in a single class need have the same format; "variant records" (in Hoare's terminology) may be specified, and then the setting and checking of a type field in each node can be taken care of automatically to identify which logical partition of the node's space is currently in effect.

Finally, the programmer specifies the names of any auxiliary operations he wants generated (initialization, copying of all of one node's fields to another node), and gives names for any variables he wants to monitor himself (number of nodes allocated, space remaining against the current limit if variable)--if omitted hidden locations are used.

As will be evident, this mechanism contains a great deal of flexibility. We specify in the same compact format everything from a bit array up to a hashed symbol table with dozens of attributes kept in related areas on different types of storage.

There are several advantages in such a flexible system. First, of course, is simplicity; a programmer's brief logical template may in complex cases generate hundreds of lines of intricately-optimized assembly code without detailed specification. This, in turn, makes it easy to enforce Project coding standards; it is so much simpler to do things the right way than to misuse any other mechanism. We also have standard documentation of formats as an automatic by-product. Thus, changes such as adding fields, or changing their sizes, can be done by anyone in a few minutes whether or not he knows the code to use the data-definition he is altering.

Such standardization also pays off in that we can write "generic", or "polymorphic", procedures in our Gasp language which can work with any node collection on any machine. For instance, we have a standard sorting routine which sorts any table (any contiguous sub-range of an allocation area) on an integer field, without regard to specific data layout. Or, we have a standard binary-search routine which searches any table sorted in that way on a field. Considerable attention was paid to these routines (alternative versions were coded, timed and checked out carefully) and now only one version needs to be maintained in libraries or loaded into core.

Like the Gasp translator, the data-definition compilers can be operated in debug-mode as an option set by toggle; in this mode they generate code to check at run-time everything that can be reasonably checked-- not only that a proffered pointer is in the proper collection, but that the actual index is below the stack top (or "high water" for a free list), that a value to be inserted will fit into the field available for it, that a variant record-type is used correctly, etc. Such full run-time checking can expose mechanical errors in programs at a very early stage of development rather quickly, at some cost in space and time; but all the checking code (except bounds checking) can be omitted at any time by simply re-compiling in non-debug mode.

A key to making this system operate efficiently is that the data-definition compilers work very hard to optimize special cases--taking advantage of field alignment to suppress redundant instructions, compiling all possible constants directly into instructions, and the like. The CDC 6000 compiler goes to extraordinary lengths to make good use of "Extended Core Storage" (ECS), setting up paging buffers in main core with clean/dirty bits, and being careful not to incur the penalties of re-fetching when unnecessary.

All this optimization of packed-data would be impractical to carry out by hand repeatedly; but once done in our system, all new definitions or changed definitions are automatically given a re-optimization to take advantage of any economies possible.

6. Quince System Report Program Generator

As a good example of the sort of savings possible with such standardized data-definitions, there is the program-writer which has been implemented to produce listings. It accepts a few lines of information about the titles and subtitles, column headings and spacing, and which fields of which tables are wanted (including pointer following). It generates a Gasp program to properly format all numeric and character fields into listings, automatically making multi-page widths as required.

This answers a very practical problem; we want our system to run entirely without a Fortran format-interpreter--not only because of the notorious inefficiency of most format processors, but because we have special defined formats of our own to consider and because Fortran format specifications are a particularly vexing source of machine-dependencies, and we fairly often have character-strings to handle at the edges of the machine-translation system. The labor of constructing a program by hand to convert and format disparate types is, however, so great that our programmers would scarcely have time or energy to carry it out. Fortunately, it is unnecessary, since definitions of all the data formats are available for generating the programs mechanically; the specification is thus actually briefer than writing Fortran with formats would be. Better still, a report specification written once is directly usable on all machines, since its machine-dependencies are lowered into the storage-mapping procedures.

Again, we encourage conformance to system standards; the facility is so easy to use (a report-definition can be dropped into a Gasp program at any point) that there is no reason to use anything else, and thus all our listings emerge titled and formatted in a standard form so that anyone can read another's output secure that he knows the ground rules (never a question, "is that octal or decimal?"), and that no mechanical coding errors have intervened to print part-fields, say, and that all error conditions have been checked properly.

7. Quince System Storage Definition and Initialization

For intra-procedure definition of storage, we have implemented a program which handles all the tasks which would ordinarily be handled by Fortran COMMON and EQUIVALENCE definitions--those most notoriously inflexible and error-prone statements.

To use this program, a programmer codes lines specifying first a "block" name and a size (to become a common block) and then for each variable and "area" (of the sort used for node storage) codes a single line giving its symbolic name, size in words, type (integer, etc.) initial value if any, and a brief comment on significance. Finally, he gives names for macros which can include information to be generated about several logically-related variables and areas.

The program reading this generates a Fortran common block definition, and EQUIVALENCE declarations equivalencing the symbolic names to locations in the block. Any necessary dimension or type declarations are added (checking the Fortran spelling rules) and a comment is included containing the description provided by the programmer.

These are written out as macro bodies, so the programmer now has available a large number of macros which he can invoke with a single name which will drop into his code a comment explaining a symbolic name, the equivalence of that to a common location, and all necessary declarations--or such information about a group of names can be in a single macro.

The program also generates a parallel specification--comment, space, offset--in either CDC or IBM assembler, and makes it into another macro-body under a systematically-related name.

If any variables are to have initial values the program generates a Fortran block data (or several, if many values are involved)--these can get quite lengthy sometimes with all the necessary declarations and no multiple-definition facility--and writes out that.

Finally, the program generates a standard format for documentation which shows the programmer's input and where each item has been located, together with a note on unused space in the block.

To use these results in his routines, a programmer simply writes code in Gasp using the symbolic names, and invokes the appropriate definition macro at the head of the routine. If storage assignments change, the macros will be re-generated by re-running the storage-definer, and then the affected routines will need to be recompiled; but nothing in the code will need to be manually adjusted.

The fact that each routine typically uses only a few symbolic names, and has definitions only for those it uses, means that we avoid over-burdening any Fortran compiler's symbol table with deadwood and prevent inadvertent collisions between local and global names.

The program to carry out this process is exceedingly trivial, but has given very satisfying results. Vast amounts of clerical work are taken over by the program, and problems just never occur from varying storage definitions among Fortran routines, or between Fortran and assembler definitions, and none of the mysterious problems caused by keypunch errors in these crucial communication statements. It is a sobering thought to reflect that, in essence, this is the old Jovial COMPOOL mechanism, somewhat elaborated. We have made haste very slowly in tools for systems programming.

8. Quince System Optimization and Monitoring Techniques

Considerable effort has been expended in providing debugging and error-handling facilities, and tracing and timing information about the system. Some of the facilities are implemented only for the "research version" of the system, in CDC 6000 code--timing, symbolic names in statistics, and automatic re-initialization after catastrophes.

All errors in the system go to a single error-handler, bearing (1) the calling routine's identification, (2) the error number within that routine assigned by the programmer (or for debugging code inserted by the Gasp translator or data-definition compilers, a generated error number), (3) a severity code between 1 and 100, and (4) one word of optional extra (variable) information.

The error handler writes this information to an error log, and then begins crawling through the code following the call chain (this is not a facility provided by the CDC operating system, so considerable ingenuity was required to extract this information from both Fortran and assembler machine code) and finding all the parameters at every call level. As the error-handler hauls itself up the call chain, it writes to the error log (beginning with its caller) the symbolic name of the routine, the location of the call, the addresses of all parameters, the values of the parameters (in characters, integer, and octal), the number of parameters prepared and passed, and the number of parameters expected; this is repeated for every level up to the main program.

Finally, the severity code is checked against a threshold for the accumulated control is passed to a "final statistics" module which will add tally and time information to the error log and end the run; otherwise the error-total is incremented, and control is returned to the caller.

Each individual routine in the system can individually participate or not in tallying--incrementing a counter by one each time its entry point is used (very cheap)--and timing--a real elapsed-time accounting of the milliseconds spent in the routine (not a sample--rather expensive).

At the end of a run this information is formatted and printed; for each routine called during the run, its symbolic name is listed, the tally of entries, the time in milliseconds, and a histogram bar-graph is printed alongside the names showing the time distribution. All this appears whether a run is ended normally by the programmer or terminated forcibly by the error-handler.

When debugging code is generated by the Gasp translator, it is virtually impossible for a test run to end uninformatively; every loop in the routines with such code has an extra "loop counter" tested every time, generating an error call after a certain parameter count is exceeded--without go-to's, it is impossible to enter a loop without incrementing the counter. Because the storage-mapping procedures will not write outside the areas they are charged with, it is virtually impossible to write data over instructions. Most conditions which could cause operating system intervention are checked either at translation time or at run-time in debug mode; the debug output printer even double-checks to be sure it is not printing so many lines of diagnostic information that it will get cut for that reason.

Thus, a programmer can have considerable confidence that a run will end with at least a comprehensive error log and full run tallies and timings, providing easily-used information. It is never necessary to pore over core dumps hypothesizing what happened a million program steps back.

This is "heavy-duty" debugging power, and it has amply paid for itself. Carefully written "structured" programs are most apt to suffer from mechanical bugs (keypunch errors in names, for instance) which produces mysterious and non-diagnostic behavior, rather than errors in logical control flow. Since our monitor system reports unusual behavior early, with contemporaneous reports on contents of crucial cells, these become much easier to fix. In conjunction with the help of the program-generators in avoiding clerical errors and mis-match errors, it is quite usual for fairly complex modules to be checked out in isolation in one to three tests; the parameter checking is even more useful in then integrating such modules. This speeds up production, and tends to make it predictable, in terms of both time and money.

Best of all, of-course, all the cost of the monitoring and debugging system can be removed selectively by purely mechanical means. Each Gasp routine and each storage-mapping definition can have its debugging code omitted from a new compilation by merely setting a toggle; independently the timing analysis code can be omitted, as can the tally code.

The natural path is to begin omitting these from the bottom up; the timing code is constructed so that any un-timed routine has its time charged to its caller (or to its caller, etc.), and the two kinds of routines can freely be intermixed with timed routines surrounding an untimed one in the call pattern; this retains the value of the figures for over-all analysis.

Ultimately, of course, a group of modules migrates toward the status of a "production system" as debugging code is completely removed, leaving only the irreducible minimum of run-time overhead. Since this migration is accomplished by program rather than by hand, there is no chance of accidentally introducing random variations between a production system and its monitored predecessor.

One of the best uses of the timing analyses is to aid in selective optimization. It is useless, of course, to shave nanoseconds off a routine called once per run. Quite early in the development of the new system, however, we employed such timing analysis on the first module, completed more or less as a demonstration of the technique to be used; a simple change in the heavily-used routines speeded the module up to 46% of its previous execution time.

It is not yet possible to be sure how fast the first release of the new system will run but there is some reassurance that whatever its performance we will know where to expend effort to improve it. We will know where the whole system is spending its time, and so know where either code-tightening or algorithm changes are worth considering. To aid in this analysis, a standard routine callable at any point will log a timing analysis for a programmer and optionally re-initialize everything. We have already used this strategy on individual routines and modules, comparing relative timings of alternative strategies, so it is ready for use on the integrated system as well.

9. Evaluation of Quince System Implementation

As was mentioned at the outset, none of the system implementation tools or techniques is, in itself, a new contribution to the state of the computing art; each has an academic history, and a number of precursors, and recently it has seemed that every issue of the computer journals has contained a prospectus for similar work [6,7,16,40].

What is more unusual is the integration of a whole range of programming tools, the length of time that we have been using them, and the size of the programs produced with their aid.

Below a certain "critical mass", such programming tools remain no more than Fortran coding aids. Used in this way they offer a welcome help in abridging the clerical work of coding; but coding is a rather small part of the programming problem in a system as complex as the Quince system, and coding aids provide little help in the more difficult tasks of program conception, design, and verification [5,9,13,17,31,48]. There have been several commercial organizations, for instance, which have found it economical to use programs to read storage allocation definitions and punch Fortran COMMON and EQUIVALENCE cards, which are then manually interspersed in Fortran program units [15]. What is basically the same scheme takes on a quite different appearance in the Quince system, when the result is macro bodies rather than cards, and when the allocation is made available to the data accessing system; with these extensions, the macro invocations in the source code have the same conceptual status as declarations, the more so since the programmer maintains the storage definition at the allocator-source level (from which changes propagate automatically into macro bodies, and thence into procedures invisibly).

Similarly, there are by now any number of people doing simple translation of go-to free control structures into Fortran, PL/I, APL [26], 360 Assembler, etc., etc. (a really welcome coding aid since many programmers now perform this tedious step by hand) [31]. The idea becomes much more helpful when combined with conditional insertion of debugging and monitoring code, and a syntax for invoking the data creation and accessing procedures and for triggering the allocation insertion.

The length of time that we have had to experiment with our programming tools is quite unusually long. Although a few (mostly oversimple) precompilers for adding control structures to Fortran are now emerging onto the market, we have been using our translator for more than two years, and the rest of the system for almost two years. In that time we have used it to produce around 50,000 lines of system-language source code for the Quince system -- a large programming project of much more than usual complexity. This is a fair length of time and a fair test for the approach.

Our experience with these methods of software production has been so favorable, surpassing even our own predictions, that we cannot now imagine reverting to the formerly conventional methods.

During the present contract we have produced just about ten times as much executable code for the Quince machine translation system as was produced for the SAS during our previous contractual efforts; and the new code is much easier to maintain, much more reliable, and portable as well.

This productivity is in addition to the work on the programming tools themselves, which indicates (correctly) that even good tools need not take too long to produce. In our case, we invested the time of three people for about four months; however disquieting this investment was at the beginning of the contract; it had been recouped with a handsome profit before the contract's end, and leaves us in a stronger position for future work as well.

Nor have the computer costs of the Quince techniques been burdensome. Before beginning, we took into account the possibility that constant translation from our source formats for procedures and data definitions might be so large that we would have to maintain copies of our system at the intermediate (Fortran and Assembler) level [23]. That turned out not to be a worry. Our translators are so much faster than Fortran compilers and assemblers that we have a net saving in computer time because of fewer debugging and testing runs. Hence, we have been able to maintain the system entirely in our own source code, without the need to pay particular attention to the fact that the intermediate Fortran representation is produced.

In sum, the whole Quince system development has been a notable success. We appear to have multiplied our ability to produce quality software by at least a factor of ten. This will permit us to continue to improve our current capabilities and to experiment with adding completely new capabilities at a much quicker tempo than has been possible in the past.

References

1. Aho, Alfred V., and Jeffrey D. Ullman. The Theory of Parsing, Translation, and Compiling. Englewood Cliffs, N.J.: Prentice-Hall. Vol. I: Parsing, 1972; Vol. II: Compiling, 1973; 1002 pp.

2. American National Standards Institute. American National Standard FORTRAN X3.9-1966.

3. ANSI Subcommittee X3J3. "Clarification of FORTRAN Standards--Initial Progress." Communications of the Association for Computing Machinery 12 (May, 1969), 289-294.

4. ANSI Subcommittee X3J3. "Clarification of FORTRAN Standards--Second Report." Communications of the Association for Computing Machinery 14 (October, 1971), 628-642.

5. Aron, Joel D. The Program Development Progress, part I. [The Systems Programming Series.] Reading, Mass.: Addison-Wesley, 1974. 264 pp.

6. Bauer, F. L., and J. Eickel, eds. Compiler Construction; An Advanced Course. [Lecture Notes in Computer Science 21.] Berlin: Springer, 1974. 621 pp.

7. Bauer, F. L., ed. Advanced Course on Software Engineering. Berlin: Springer, 1973. 545 pp.

8. Belady, L. A., and M. M. Lehman. "Programming System Dynamics; or, The Meta-Dynamics of Systems in Maintenance and Growth." IBM Research Report RC 3546 (#16021). September, 1971. 30 pp.

9. Brooks, Frederick P., Jr. *The Mythical Man-Month: Essays on Software*. Reading, Mass.: Addison-Wesley, 1975. 195 pp.

10. Brown, P. J. *Macro Processors and Techniques for Portable Software*. London: John Wiley and Sons, 1974. 244 pp.

11. Clark, Brian L. "The Design of a System Programming Language." [Project SUE System Language.] M.S. Thesis, Department of Computer Science, University of Toronto, November 1971. 62 pp.

12. Cocke, John, and J. T. Schwartz. "Programming Languages and Their Compilers, Preliminary Notes," 2nd ed. Courant Institute of Mathematical Sciences. New York University, 1970. 767 pp.

13. Dahl, O.-J., E. W. Dijkstra, and C. A. R. Hoare. *Structured Programming*. [A.P.I.C. Studies in Data Processing 8.] London: Academic Press, 1972. 220 pp.

14. Day, A. Colin. *Fortran Techniques with Special Reference to Non-Numerical Applications*. Cambridge: Cambridge University Press, 1972. 96 pp.

15. DeSalvio, A. J., J. G. Purdy, and J. Rau. "Creation and Control of Internal Data Bases Under a Fortran Programming Environment." Communications of the Association for Computing Machinery 13 (April, 1970), 211-215.

16. Desjardins, Pierre. "Dynamic Data Structure Mapping." Software--Practice and Experience 4 (1974), 155-162.

17. Dijkstra, Edsger W. "A Constructive Approach to the Problem of Program Correctness." BIT 8:3 (1968), 174-186,

18. Fang, Isu. "Folds: A Declarative Formal Language Definition System." Stanford University Ph.D. Thesis. Stanford Computer Science Department Report No. STAN-CS-72-329, 1972. 286 pp.

19. Graham, Robert M. "Use of High Level Languages for Systems Programming." Project MAC, Programming Linguistics Group Memo No. 2. Massachusetts Institute of Technology: November, 1969. 18 pp.

20. Habermann, A. N. "Critical Comments on the Programming Language Pascal." Acta Informatica 3:1 (1973), 47-57.

21. Hall, A. D., et al. Altran Reference Manual. Bell Telephone Laboratories, September, 1972.

22. Hetzel, William C., ed. Program Test Methods. Englewood Cliffs, N.J.: Prentice Hall, 1973. 328 pp.

23. Hoare, C. A. R. "Hints on Programming Language Design." Stanford University Artificial Intelligence Laboratory Memo AIM-224, Report No. STAN-CS-73-403, December, 1973. 29 pp.

24. Irons, Edgar T. "A Syntax-Directed Compiler for Algol 60." Communications of the Association for Computing Machinery 4 (1961), 51-55.

25. Jensen, Kathleen, and Niklaus Wirth. Pascal: User Manual and Report. [Lecture Notes in Computer Science 18.] Berlin: Springer, 1974. 170 pp.

26. Kelley, R. A. "APLGOL, an Experimental Structured Programming Language." IBM Journal of Research and Development (January, 1973), 69-73.

27. Kernighan, Brian W., and P. J. Plauger. The Elements of Programming Style. New York: McGraw-Hill, 1974. 147 pp.

28. Knuth, Donald E. "Semantics of Context-Free Languages." Mathematical Systems Theory 2:2 (1968), 127-145. Correction in Mathematical Systems Theory 5:1 (1971), 95-96.

29. Knuth, Donald E. "Examples of Formal Semantics." In E. Engeler, ed., Symposium on Semantics of Algorithmic Languages. [Lecture Notes in Mathematics, 188.] Berlin: Springer, 1971, pp. 212-235.

30. Knuth, Donald E. "A Review of Structured Programming." Stanford University Computer Science Department, Report No. STAN-CS-73-371, June, 1973. 25 pp.

31. Knuth, Donald E. "Structured Programming with goto Statements." Computing Surveys 6:4 (December, 1974), 261-301.

32. Larmouth, J. "Serious Fortran." Software--Practice and Experience 3 (1973), 87-107 [part I], 197-225 [part II].

33. Larson, Chris. "The Efficient Use of Fortran." Datamation (August 1, 1971), 24-31.

34. McCracken, Daniel D., and Gerald M. Weinberg. "How to Write a Readable Fortran Program." Datamation (October, 1972), 73-77.

35. Ogdin, Jerry L. "Designing Reliable Software." Datamation (July, 1972), 71-78.

36. Parnas, D. L. "On the Criteria to be Used for Decomposing Systems into Modules." Communications of the Association for Computing Machinery 15 (December, 1972), 1053-1058.

37. Parnas, D. L. "A Technique for Software Module Specification with Examples." Communications of the Association for Computing Machinery 15 (May, 1972), 330-336.

38. Perstein, Millard H. "The Jovial (J3) Grammar and Lexicon." System Development Corporation Technical Memorandum TM-555/002/04(AB), May, 1966. 138 pp.

39. Petrick, S. R. "On the Use of Syntax-Based Translators for Symbolic and Algebraic Manipulation." In S. R. Petrick, ed., Proceedings of the Second Symposium on Symbolic and Algebraic Manipulation. New York: Association for Computing Machinery, 1971, pp. 224-237.

40. Robinet, B., ed. Programming Symposium; Proceedings, Colloque sur la Programmation, Paris, April 9-11, 1974. [Lecture Notes in Computer Science, 19.] Berlin: Springer, 1974. 425 pp.

41. Satterthwaite, E. "Debugging Tools for High Level Languages." University of Newcastle-upon-Tyne, Computing Laboratory, Technical Report Series No. 29, 1971. 32 pp. Also, Software--Practice and Experience 2 (1972), 197-217.

42. Simmons, Dick B. "The Art of Writing Large Programs." I.E.E.E. Computer (April, 1972), 43-49.

43. Ullman, Jeffrey D. "Applications of Language Theory to Compiler Design." In Alfred V. Aho, ed., Currents in the Theory of Computing. Englewood Cliffs, N.J.: Prentice-Hall, 1973, pp. 173-218.

44. Vyssotsky, V. A. "Common Sense in Designing Testable Software." In William C. Hetzel, ed., Program Test Methods. Englewood Cliffs, N.J.: Prentice-Hall, 1973, pp.41-47.

45. Waite, William M. Implementing Software for Non-Numeric Applications. Englewood Cliffs, N.J.: Prentice-Hall, 1973, 509 pp.

46. Wilner, Wayne T. "Formal Semantic Definition Using Synthesized and Inherited Attributes." In Randall Rustin, ed., Formal Semantics of Programming Languages. Englewood Cliffs, N.J.: Prentice-Hall, 1972, pp. 25-40.

47. Wirth, Niklaus. Systematic Programming. Englewood Cliffs, N.J.: Prentice-Hall, 1973. 167 pp.

48. Wirth, Niklaus. "On the Composition of Well-Structured Programs." Computing Surveys 6:4 (December, 1974), 247-259.

III. LEXICOGRAPHY AND TEXT ORGANIZATION

1. Accession of New Dictionary Entries

During the contract period, a total of 15,930 new dictionary entries in the field of nuclear physics (with concentration in the areas of material structure, reactor engineering and technology, nucleonics, and nuclear and thermonuclear reactions) and related fields (mathematics, engineering, etc.) were processed and added to CHIDIC, increasing the total number of entries in this dictionary file to 82,330.

Dictionary entries were obtained from two types of sources:

1. Chinese nuclear physics texts, which were studied sentence by sentence, consulting system sentence dictionary lookups (single-direction [left-to-right] or double-direction [left-to-right and right-to-left] lookups) for each sentence in various selected portions of the texts listed below. In some cases a more comprehensive lookup approach was employed; this consisted of simultaneously performing right-to-left and left-to-right lookups, 'minimum-span' lookup (whose result is the set of sentence terminals covering the sentence from beginning to end in the minimum number of lexical terminals or 'spans' [equivalent to 'longest-match' lookup]), and 'full' lookup (which provides all possible sets of sentence terminals, i.e., minimum spans + 1, minimum spans + 2, etc.). The system dictionary lookups were checked against the content of each text sentence to determine

adequacy and appropriateness of coverage, and prospective additions to the dictionary were noted. Texts so processed included the following:

β he γ Fang-she-xing Zhuan-ji (Collected Papers on β and γ Radiation), Mei Zhen-yue, et al.; Science Publishers, Peking, October 1963.

Yuan-zi-he Neng Gai-lun (Introduction to Nuclear Energy), Tang Shang-song, et al.; Chinese Culture Publishing Enterprises Commission, Tai-pei, August 1957.

Yuan-zi-he Neng (Nuclear Energy); Cheng Zhi-zhong, ed.; Science and Technology Publishers, Shanghai, December 1957.

Yuan-zi Neng de Yuan-li he Ying-yong (Principles and Applications of Atomic Energy); Zhao Zhong-yao, et al.; Science Publishers, Peking, July 1965.

Yuan-zi he Yuan-zi Neng (Atoms and Atomic Energy); He Ji-mei; Educational Illustrations Publishers (location and date of publication not specified).

Fan-ying-dui Wu-li-xue yu Yan-jiu-yong Fan-ying-dui (Reactor Physics and Research Reactors), Volumes 1,3, and 4; compiled by Editorial Board, Committee on Nuclear Science, Chinese Academy of Sciences; Science Publishers, Peking, April 1961.

Shou-kong Re-he Fan-ying (Li-lun Ji-chu ji Tan-suo Cheng-jiu) (Controlled Thermonuclear Reactions [Theoretical Foundations and Research Achievements]); Lu He-fu, et al.; Shanghai Science and Technology Publishers, Shanghai, August 1962.

Ji-ben Li-zi (Fundamental Particles); Yang Zhen-ning, et al.; Zhong-hua Publishers, Taipei, March 1967.

Zhai-yi ji Guang-yi Xiang-dui-lun (Special and General Theories of Relativity); Wu Da-xian; Zhong-hua Publishers, Taipei, March 1968.

2. Bilingual Chinese-English and English-Chinese scientific and technical dictionaries, primarily the following:

Ying-Han Yuan-zi-he Ci-dian (English-Chinese Dictionary of Nuclear Terminology); translated by Editorial Board, Committee on Nuclear Science, Chinese Academy of Sciences; Chinese Industry Publishers, Peking, October 1963.

Chinese-English Dictionary of Nuclear Physics Terminology; compiled by the Foreign Technology Division, Wright-Patterson Air Force Base; undated.

Han-Ying Ke-ji Chang-yong Ci-hui (Chinese-English Dictionary of Common Scientific and Technical Terminology); Translations Office, Chinese Academy of Sciences; Commercial Printers, Hong Kong branch; July 1971.

Prospective dictionary entries from texts or dictionaries were subjected to the following steps of treatment or processing:

1. Checking for relevance to the context of nuclear physics in particular and overall scientific-technical context in general;
2. Checking adequacy and appropriateness of English gloss;
3. Assigning grammar code, into which is incorporated as much retrievable information as possible regarding the syntactic, semantic, and other features of the entry;
4. Assigning Standard Telegraphic Code numbers to represent the Chinese characters;
5. Keypunching.

2. Random-Access Dictionary

Development of the random-access disc dictionary (described in detail in an earlier report [see RADC-TR-74-22]) has been discontinued, the reasons being that (1) there is no suitable disc hardware available locally to support further development and testing (the University's Computer Center being several years behind schedule in making appropriate disc facilities available), and (2) during the present contract period the Foreign Technology Division, Wright-Patterson Air Force Base, withdrew its requirement for the initial-production-capability Chinese-English machine translation system of which the random-access disc dictionary was a component, and the resultant reshuffling of the Project's priorities necessitated indefinite postponement of work on the disc dictionary.

It is to be hoped that we will be able to resume development of the random-access disc dictionary at some time in the not too far distant future; its completion would greatly enhance the efficiency of the overall system.

3. Concentric Dictionary Organization

The Project now maintains and utilizes dictionary files on three different levels for different purposes. Each level is discussed briefly below.

1. CHIDIC, which for a number of years has served as our all-inclusive dictionary file, has now become a more-or-less inactive repository of machine-readable dictionary entries, many of which are not considered applicable in the context of nuclear physics or general scientific and technical writing in Chinese. The CHIDIC file remains of considerable utility, however, as a data base for linguistic and lexicographic research, and it is hoped that in the future the CHIDIC file might be enlarged by merging it with the McGraw-Hill dictionary and the Department of Defense dictionary, which would furnish an even larger data base for research (see section 5).

2. PHYDIC (for PHYSICS DICTIONARY), which at present contains approximately 35,000 entries, was created from its predecessor, CHIDIC, by selecting all lexical items occurring in our half-million-word corpus of Chinese nuclear physics text; it also includes other select items deemed of relevance in general scientific and technical writing in Chinese. PHYDIC is the largest dictionary file presently utilized in processing the

text submitted to the Syntax Analysis System for translation, and it is toward PHYDIC that dictionary maintenance and updating effort is currently directed. Reducing the scope of our day-to-day dictionary efforts by limiting the coverage of the dictionary to what we may reasonably expect to encounter in the course of processing parts of our nuclear physics text corpus has made both maintenance and updating of the dictionary more manageable and has sharpened the focus of the Project's on-going linguistic re-evaluation of the dictionary's content. Linguistic and lexicographic work in particular should be able to proceed at a somewhat faster pace.

3. Subdictionaries: Due to the fact that PHYDIC in its entirety far exceeds the system capacity with regard to dictionary size, as well as to the fact that work on the random-access dictionary component has been halted (see above), we continue to find it necessary to select a subdictionary from PHYDIC each time an individual text is to be processed. Contrary to our earlier procedure, however, these subdictionaries are no longer being updated; as was just mentioned above, all dictionary updates are done at the level of PHYDIC. This necessitates the selection of a new subdictionary after each dictionary update, but it also obviates the need of constant bookkeeping to keep track of all dictionary update transactions to ensure that they ultimately filter back into the highest-level dictionary (PHYDIC).

4. Dictionary Updating and Maintenance

Since all dictionary updating is now performed at the level of

PHYDIC, it was necessary that certain complex and time-consuming dictionary updating procedures be modified to make the whole task of dictionary updating more streamlined and economical. This was also necessitated by the demands of the Quince system, which required conformance on the part of the dictionary to new data types. To this end, a dictionary maintenance package was designed with several objectives in mind: simplicity, flexibility and ease of use; low cost per run (especially important in consideration of the much larger size of the PHYDIC file compared with the average subdictionary); reliability; and ease of program maintenance.

Reducing the cost factor as much as possible was important for yet another reason, namely, that low cost permits more frequent updating, which in turn reduces the 'linguistic turn-around', the time it takes for changes conceived of by the lexicographer or linguist to be implemented in the dictionary via updating. This seemingly trivial matter actually assumes a fair amount of importance in consideration of the size of the dictionary and of the large numbers of changes being initiated by three or four linguists at once. The linguists' time must be devoted as much as possible to linguistics and not to keeping track of a mounting pile of changes from update to update.

Since the programmers' time is just as valuable as that of the linguists, it was clear that a dictionary maintenance package so designed as to permit the linguist or lexicographer to update the dictionary at any time without having to trouble the programmers would be a great benefit, and would substantially increase the efficiency and manageability of the dictionary updating process.

The dictionary maintenance package designed on the basis of these and other considerations has proven itself admirably in use, and it provides a high degree of flexibility through the incorporation into the package of a number of options, including automatic sorting of the dictionary into order by grammar code, automatic selection of a subdictionary for a specified text, printing of only a specified part of the whole dictionary file, and automatic assignment of romanized pronunciation (in any one of four romanization systems: pin-yin, National or 'Guo-yeu', Wade-Giles, or Yale) via table look-up immediately prior to output printing (thus making it unnecessary to devote space in each record to carrying pronunciations).

The dictionary maintenance package recognizes dictionaries in two different formats and sizes, one of 80-column records and the other of expanded 132-column records. The expanded format, generated by a package option, is the format which fits the design specifications of the Quince system; the smaller 80-column format is suited to frequent and heavy updating.

The package currently contains three programs, SCANNER, UPDATE, and SELECT, all of which are relatively small, straightforward, and functionally organized, with sufficient comments to make program maintenance and modification relatively easy.

SCANNER reads option cards and transactions for each run, producing a listing of transactions and flagging and deleting transactions containing obvious errors (see Figure 3-1).

```
   D 02553333                  NB
   C 02553333                  VI          GVNTHB
   C 02553333                  VI          ESTOP+UP
   C 02553966                  VI          ESTAY
*** ERROR - UNRECOGNIZABLE CARD ***
 → 2 02557319                  VI           STOP
   D 02782676                  VI
   D 03423296                  VI
   D 03970606                  DP
   D 03970606                  NB
   C 03970606                  VI          ECOOL
*** ERROR - BLANK ENGLISH GLOSS ***
 → I 00071071                  VIN
   C 00081186                  VI          GVA
   I 2521034235642494          NB           DAYLIGHT+ILLUMINATION
   I 2521703514104453          NB           DAYLIGHT+RANGE
*** ERROR - INVALID CHANGE CARD ***
 → C 00015268                  VQS
*** ERROR - INVALID TELECODE ***
 → D 000162
*** ERROR - INVALID CHANGE CARD ***
 → C 000528174619              NN
   D 71436670                  NN
   D 89558641055306783670      NZ
```

Figure 3-1. Scanner Output Sample

UPDATE, which reads the dictionary and a sorted, edited transaction file, searches the dictionary sequentially, carries out the transactions, and writes a new dictionary. It also provides a listing of all transactions performed (see Figure 3-2).

SELECT produces a subdictionary from a dictionary by passing the dictionary against a list of search goals of a maximum of four and one half telecodes in length, selecting for inclusion in the subdictionary each entry which matches a search goal.

```
INSERTION: NEW ENTRY
     I 51746852                      NB        * ENERGY

INSERTION: NEW ENTRY
     L 54783044                      1D-UM     * FAHRENHEIT

CHANGE:    ORIGINAL ENTRY
     J 59440354                      VTH       - LOAD+INTO
CHANGE(G): REVISED ENTRY
     I 59440354                      VNTHB/NN  LOAD+INTO

INSERTION: NEW ENTRY
     3 5944035411208199              VTHB/NN   ENCASE

INSERTION: NEW ENTRY
     M 59443240                      VTH/NN    ADD*LOAD

CHANGE:    ORIGINAL ENTRY
     J 59446080                      VTH       INSTALL
CHANGE(G): REVISED ENTRY
     I 59446080                      VTH/NN    INSTALL

INSERTION: NEW ENTRY
     Y 5944608003080892              NN        INSTRUMENTATION

DELETION:  ENTRY DELETED
     J 59446792                      VTH       ASSEMBLY

CHANGE:    ORIGINAL ENTRY
     N 59446792                      VTH/NNS   ASSEMBLE
CHANGE(G): REVISED ENTRY
     I 59446792                      VNTHB/NN  ASSEMBLE

INSERTION: NEW ENTRY
     ( 5964699343493233604130960     NN        LIMONITE+CONCRETE

INSERTION: NEW ENTRY
     Q 651906241653                  NB        AXIAL+THICKNESS

INSERTION: NEW ENTRY
     I 65190686                      DP        AXIAL*LONGITUDINAL
INSERTION: REPLACING
     I 65190686                      DP        AXIAL

INSERTION: NEW ENTRY
     Q 651906861034                  NB        AXIAL+FIELD

INSERTION: NEW ENTRY
     Y 6519068669753024              NB        AXIAL+MIRROR+RATIO
```

Figure 3-2. Sample UPDATE Transaction Listing

5. Dictionary-Based Research of Chinese Morphology

Due to the lack of distinct word boundaries in written Chinese, various definitions of what might properly be considered as constituting a 'word' have been proposed and utilized by linguists over the years. Machine translation is an area where the requirement with regard to the 'clarity' of the definition of a word is a rather stringent one; that is to say, the machine translation system must be told precisely what a word is if the system is to be enabled to operate at a peak of efficiency and produce as little ambiguity as possible.

Why, it might be asked, is it necessary to delve so deeply into such things as the structures of Chinese dictionary entries? Isn't it enough simply to gather together a large pile of Chinese dictionaries, assign grammar codes to all the entries, and code them in machine-readable form? Given the present state of Chinese lexicography, we must assert that such an approach, although adequate to the task of preparing dictionaries for use by humans, is not really suited to preparing a dictionary for use by a machine, which lacks all of the knowledge of the world which the human dictionary user has available in his head. Chinese-English dictionaries have for decades suffered from a tendency toward over-lexicalization; regularities in the Chinese have been pushed aside or ignored; and very rarely is there any mention of the grammatical functions of the Chinese entries themselves -- the user of the dictionary merely makes the unconscious assumption that the grammatical identity of the string of Chinese words in a dictionary entry must be roughly the same as the grammatical identity of the string of English words given as its translation; after all, they mean the same thing, don't they?

The problem is that ordinary Chinese-English dictionaries, and especially scientific and technical dictionaries, in general do not contain the kind of reduced and clearly identified lexical building blocks which a machine translation system can make the best use of; they tend to be long lists of longer or shorter strings of words, each of which has a more or less accurate, appropriate, or idiosyncratic English gloss. And it is often the case that, because of English gloss considerations, the underlying regularities of the Chinese are hidden. In addition, most dictionaries do not contain the detailed syntactic and semantic information which is so essential in the machine translation approach. A recent notable exception is Lin Yutang's Chinese-English Dictionary of Modern Usage (The Chinese University of Hong Kong, 1972), which does identify the basic grammatical functions of its entries; the grammatical information which it furnishes, however, is no more than a drop in the 'knowledge-of-the-world' bucket.

It is important -- at least in the area of machine translation, if not generally so -- that the Chinese-English dictionary pay less attention to English and strive to more closely reflect the structural regularities of the Chinese.

Our brief forays into the jungle of Chinese morphology have been prompted by the desire to on the one hand provide the Syntax Analysis System with a useful and precise definition of a work, and on the other, by gaining a greater understanding of certain aspects of the rules which govern word (and especially noun) formation in Chinese, to bring about qualitative improvements both in the content of the lexicon itself and in the way in which the entries from the lexicon are combined by the low-level grammars during the initial stages of parsing.

The Project is currently making use of a new procedure to aid the study of certain problems related to word formation and noun compounding, a procedure which makes use of the Chinese dictionary as both a data base and an analytical tool. This is the very simple but not so obvious technique of parsing existing dictionary entries. Any given entry in the dictionary may be regarded as a text string and submitted to the Syntax Analysis System for analysis, the only extra restriction being that during the look-up phase of processing the 'dictionary-entry-as-dictionary-entry' must remain invisible to the 'dictionary-entry-as-text-string'; otherwise the 'dictionary-entry-as-text-string' would merely find itself in the dictionary during look-up (due to the principle of longest match), and nothing of interest would result. What is desired is for look-up to provide a set of sentence terminals comprised of all possible sub-parts of the original 'dictionary-entry-as-text-string' to serve as lexical input for the analysis. The system then moves on into the parsing phase, the grammars are brought to bear, and the tree-tops and English strings which result from parsing are noted and compared with the grammar code(s) and English gloss(es) of the original dictionary entry, the idea being to initially spot discrepancies between original and parse-produced grammar codes and glosses and then zero in on the actual tree structures and pinpoint areas of difficulty.

The following examples illustrate the procedure and also indicate areas where additional work is necessary.

The block of dictionary entries below serves to illustrate how the dictionary and low-level grammars may be made to function more productively to synthesize through parsing many of our present dictionary entries, thus allowing the removal of many of the longer entries from the dictionary.

1.	子体元素 1311 7555 0337 4790	NN [concrete noun] daughter+element
2.	子体物質 1311 7555 3670 6347	NN daughter+substance
3.	子体物質放射性 1311 7555 3670 6347 2397 1410 1840	NB [abstract noun] daughter+activity
4.	子体產物 1311 7555 3934 3670	NN daughter+product
5.	子体部分 1311 7555 6752 0433	NB daughter+fraction

Figure 3-3 shows the parse trees and constituents of four of these five dictionary entries. The parse tree for entry 5, not shown in the figure, is the same as those for entries 1, 2, and 4.

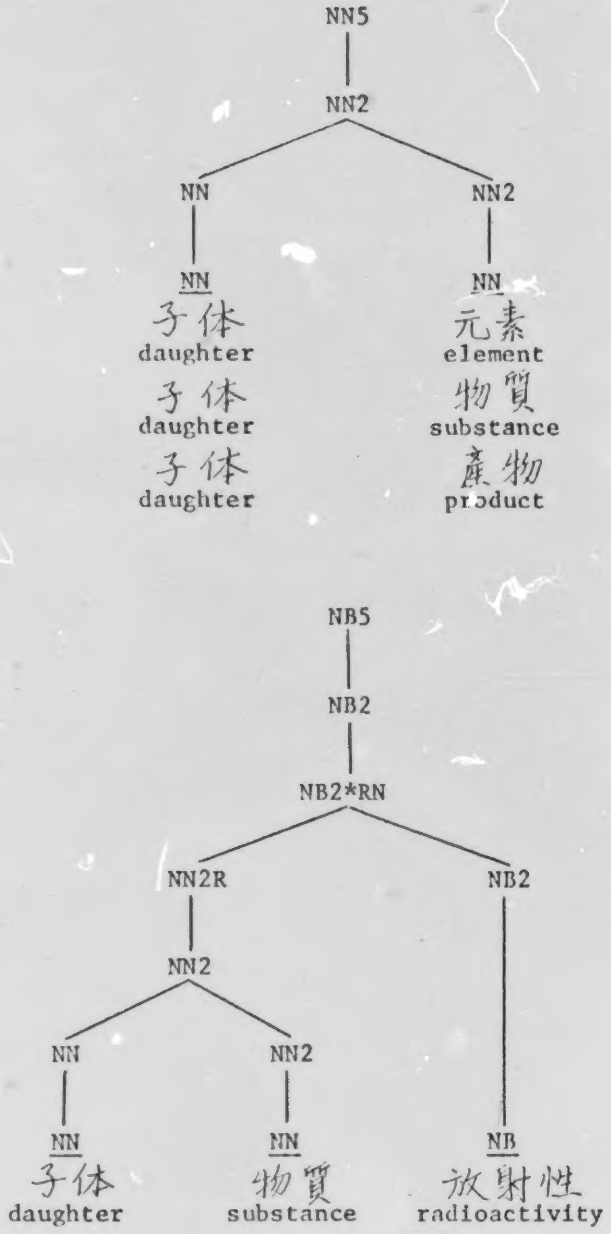
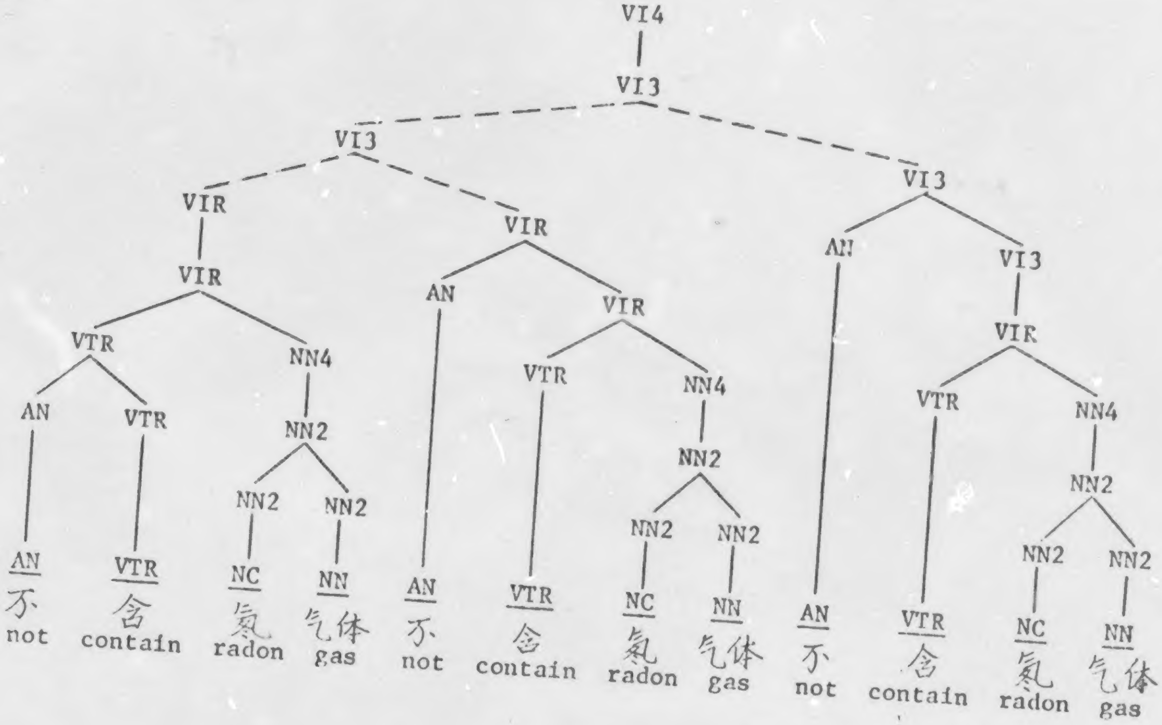

NN5
NN2
NN
NN2
NN
NN
子体
daughter
子体
daughter
子体
daughter
元素
element
物質
substance
產物
product
NB5
NB2
NB2*RN
NN2R
NB2
NN2
NN
NN2
NN
NN
NB
子体
daughter
物質
substance
放射性
radioactivity

Figure 3-3. Parsed Dictionary Entries

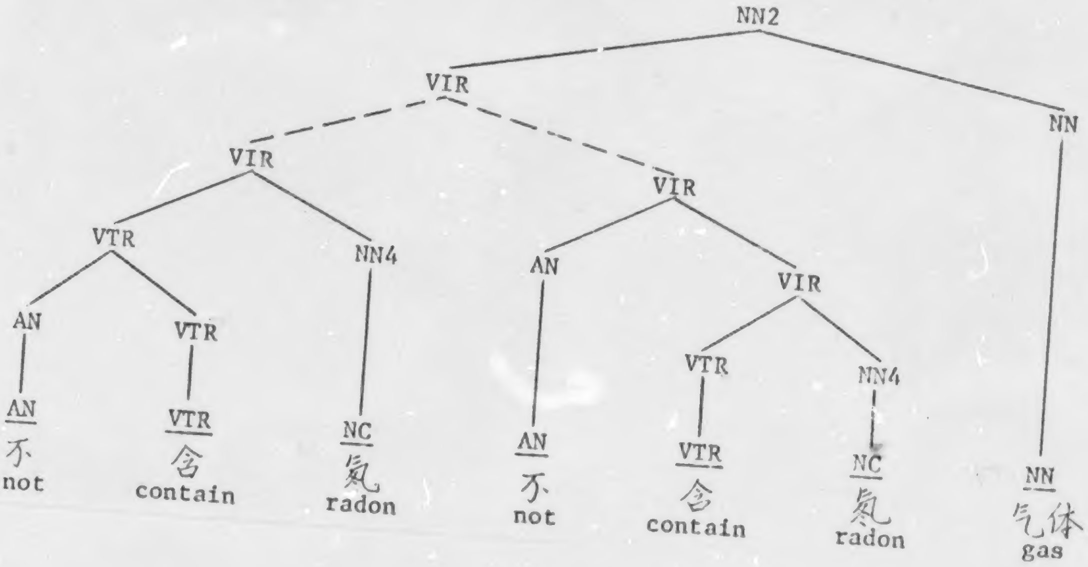
A look at the tree structures shows us the entire array of low-level word compounding rules which the system found applicable during parsing. With the actual rules used before him, the linguist can better judge whether what was claimed to be true for low-level Chinese noun phrase formation when these rules were written seems intuitively correct and of general applicability for the Chinese language in the context of nuclear physics and general scientific writing; if not, then particular changes can be readily made and checked by reparsing.

The set of examples given above resembles many other larger or smaller blocks of entries now in the dictionary which can be reduced to their component parts and need no longer be carried as longer entries. Another case is when a longer entry can not be carried due to inherent ambiguity arising from the lack of clear word segmentation. The example shown in Figure 3-4 is an entry of this type.

不 含 氡 气 体	NN [concrete noun]
0008 0698 0598 3049 7555	radon-free+gas


VI4
VI3
VI3
VI3
VIR
VIR
AN
VI3
VIR
VIR
AN
VIR
VTR
NN4
VTR
NN4
VTR
NN4
AN
VTR
NN2
NN2
NN2
NN2
NN2
NN2
NN2
NN2
NN2
AN
不
not
VTR
含
contain
NC
氡
radon
NN
气体
gas
AN
不
not
VTR
含
contain
NC
氡
radon
NN
气体
gas
AN
不
not
VTR
含
contain
NC
氡
radon
NN
气体
gas

Structure 1


NN2
VIR
NN
VIR
VIR
VTR
NN4
AN
VIR
AN
VTR
VTR
NN4
AN
不
not
VTR
含
contain
NC
氡
radon
AN
不
not
VTR
含
contain
NC
氡
radon
NN
气体
gas

Structure 2

Figure 3-4. Dictionary Entry Parse

Structure 1 has a parse-tree top node of V14; it is a predicate acceptable in the context 'X [does] not contain radon gas', where 'radon gas' is the object of 'X'. Structure 2 shows a parse as a medium-level noun phrase which is actually a relative clause in disguise; it is identical except for the absence of the relative clause marker 'de' 的, viz 'bu han dong de qi-ti' 不含氡 的 气体, and as the constituent structure shows, it has been interpreted to mean 'gas [which] [does] not contain radon'. What is required here is for the linguist or grammarian to provide a better treatment of this example, perhaps by enabling the grammar to process the string as though it were at least a potential relative clause so that the proper interlingual transformation (permutation) could be invoked to render 'not contain radon gas' into a more acceptable order, or perhaps some other solution might suggest itself. The main point here is that when the dictionary contains the long entry

不 含 氡 气 体 NN [concrete noun]
0008 0698 0598 3049 7555 'radon-free gas'

Structure 1 above, which is perfectly acceptable in a certain context, is not obtainable; because of the principle of longest-match lookup, the sub-constituents needed to obtain structure 1 would not be available as terminal input to parsing. The inherent ambiguity of this and numerous other dictionary entries is one kind of information which the technique of parsing dictionary entries makes available to the linguist, and out of better information it is hoped that better solutions to these and other problems will follow.

Initial tests of this procedure have provided some interesting results and have shown that the technique of parsing dictionary entries is useful not only with regard to investigation of Chinese morphological processes, but also appears to be of significant value in diagnosing areas of deficiency in the grammars themselves.

Henceforth the Project's lexicographic effort will be shifted from the earlier emphasis upon the acquiring of large batches of new entries for the dictionary toward providing more detailed and penetrating linguistic treatment of the dictionary relevant to the context in which we are working. Using the dictionary itself as a tool for linguistic and lexicographic research, we are beginning to focus upon a few of the long-standing major problems of Chinese morphology. Such investigation will also serve to point out areas in which overall dictionary design can be further improved so that the dictionary will be able to function more efficiently as a Quince System component and also better support the kinds of research in which it is being used.

6. Chinese Input and Output

The history of the Project's work with the problems of Chinese input is a fairly lengthy one; over the years, we have had opportunity to try a variety of hardware devices and procedures in inputting Chinese text. At the beginning, Chinese text was 'telecoded' by hand, then key-punched; later, we experimented with the Chicoder (Itek Corporation) as a Chinese input device; and three years ago we acquired the Model 600D

Chinese telecommunications equipment (see RADC-TR-74-22 for detailed description) which is now serving as our basic hardware input device. The Chinese keyboard of the 600D produces punched paper tape, each of the keyboard's 4600 Chinese and 200 non-Chinese characters being represented on the paper tape by a unique non-standard 13-bit 5-byte code. Use of this keyboard by an experienced typist has allowed us to triple or quadruple our Chinese text-coding speed, and the device is generally very easy to use and quite reliable: after coding nearly half a million words of text, we have yet to uncover one keyboard-generated error. The only problems which remain are the unique coding system generated by the keyboard to represent its characters (for our purposes, it would be much more convenient to have the keyboard generate Standard Telegraphic Code) and the paper tape itself, which is extremely inconvenient to work with with. The non-standard codes on paper tape require that two additional steps be performed after the Chinese text has been keyboarded: first, the conversion of the paper tape to magnetic tape, which must be done elsewhere, and then the conversion of the non-standard codes to Standard Telegraphic Code.

Thus, in order to eliminate these problems from our text-input operation and to further enhance our input capabilities, we have been led to consideration of how we might assemble a minicomputer/CRT device which could function as a Chinese text-editing terminal. Two principal applications are foreseen for such a device: (1) off-line use as a self-contained station for keying and editing Chinese texts of considerable length; and (2) on-line use as a Chinese computer terminal for interactive querying and updating of Chinese-language data bases.

The following is a list of those major functional capabilities which such a device should possess:

1. Chinese-character keyboard: Perhaps the only non-standard component of the system; we plan to use our present keyboard, and to make it appear as a standard peripheral to the terminal hardware by attaching a custom interface to it.

2. Chinese-character display: The operator should have a display screen for Chinese characters, either CRT or storage-tube (incremental display additions, display-destructive rewriting); fairly good resolution is required if a sufficient number of characters are to be legibly displayed at once. Chinese characters will be stored locally as vector sets, which means that some storage such as a high-performance disc must be available to hold the character vector sets, which eventually might number 15,000-20,000, although our present data base is comprised of the vector sets for around 8,000 Chinese characters (see following section).

3. Local hard-copy system: Some type of medium-quality hard-copy system must be available for 'note-keeping' of display screen content. (High-quality plotting of Chinese characters would be done elsewhere on paper or microfilm.)

4. Local editing capability: Local storage is also needed for Chinese texts of considerable length and to keep several such texts available for editing or appending additional content. Removeable local

disc storage would probably be best here as well as for storage of Chinese character vectors.

Other desirable features of such a terminal would include database transportability to large computers (via computer-compatible magnetic tape unit), conversational communication with large computers (for interactive data-base querying), and English keyboard input for mixed Chinese/English data input and for communications.

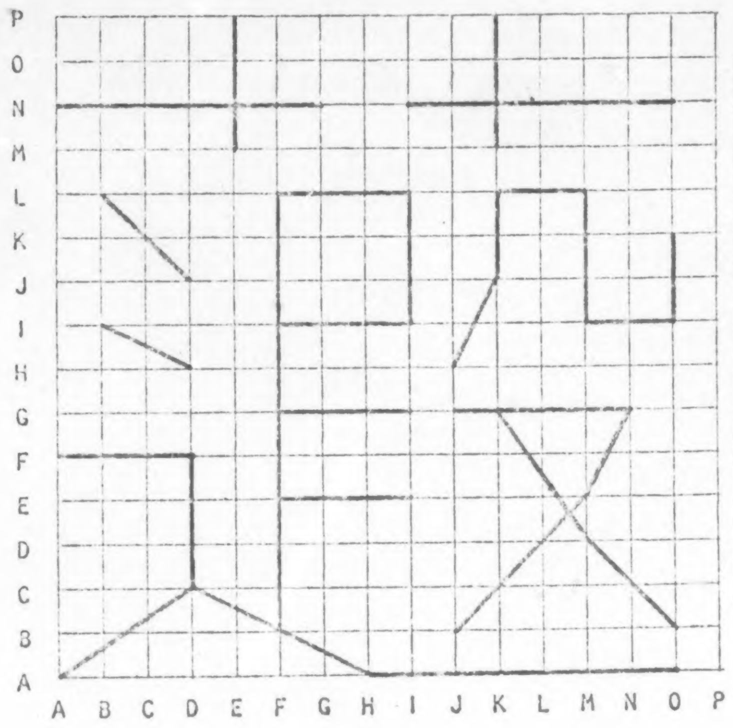
In addition to greatly increasing our text-input capability, such a terminal would provide nearly perfect hardware support for implementation of an interactive Chinese dictionary, which is clearly the way to solve some of our present difficulties in working with the dictionary itself, and allow the dictionary to become a much more accessible and useful linguistic tool.

The terminal would also be quite suitable for and greatly facilitate work with text-concordance files and sentence-feature-oriented text corpora; it would truly constitute a linguistic aid of considerable importance as well as a significant advance in the state of the art (if such it can be called) of Chinese-language input to computers.

7. Chinese Character Data Base

Much work has been done during the present contract period regarding how our data base for plotting Chinese characters might be improved and extended. We began with a tape of vector sets representing some 7,000 Chinese characters; these are the character vectors compiled under the direction of Prof. Susumo Kuno at Harvard, and are the same vectors used by CETA for plotting the Chinese characters in its dictionary. With plotting capability imminent, we were immediately faced with the need to expand the plottable font of characters to somewhere around 10,000-11,000. There seemed to be two ways to go about this task: (1) to follow Kuno's original procedure, which was to carefully and painstakingly draw each character on a 16 x 16 matrix, and then extract the vector set from the drawing, an extremely time-consuming and laborious job (see Figure 3-5); or

Step 1: character is carefully drawn on 16 x 16 matrix



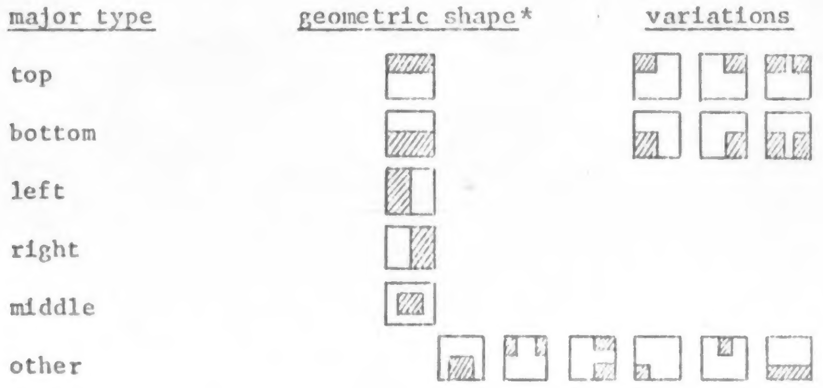
Step 2: beginning and end points of lines are hand-extracted and coded for keypunching

ANGN	EPEM	INON
KPKM	BLDJ	BIDH
AFDF	DFDC	DCAA
DCHA	HAOA	FLIL
ILII	IIFI	FLFB
FGIG	FEIE	JHKJ
KJKL	KLML	MLMI
MIOI	OIOK	JGNG
NGME	MEJB	KGMD
MDOB		

Figure 3-5. Kuno's Character Vector Coding Procedure

(2) to devise some way of composing new characters by utilizing parts of the vector sets of characters already in the plottable font. The second approach seemed much more desirable as well as much more interesting, and so it was the direction we chose to follow.

Detailed character-composition software specifications were drawn up, and programming was sub-contracted to Dynamic Graphics, Inc., a local computer graphics consulting firm with which we had had earlier experience, while Project personnel worked on problems having to do with what kinds of character-component data could readily be made available and appended to vector sets already in the data base. Chinese characters having radicals among the thirty or so most commonly-occuring radicals (out of a total of 214 radicals) were examined to determine the number of different positions which each radical might occupy in any character in which it occurred. These positions were found to comprise five major types, some of which also had several variations (see Figure 3-6).



*shaded area indicates location of radical in character

Figure 3-6. Radical Positions in Chinese Characters

Breakdowns by radical position type and frequency (i.e., number of characters with radical in position type) were obtained for these characters (see Figure 3-7).

radical	series of characters in Standard Telegraphic Code with this radical	radical location type		frequency
土 32 'earth'	0960-1109	L	◧	165
		T	⬒	3
		T_1	◰	3
		B	⬓	37

Figure 3- 7. Sample Data

Radical position data are to be appended to the character vectors and used in helping to define the dimensions and sequences of application of the various 'window' structures by which radicals, whole characters, and other character components will be assembled to form the vectors for 'new' characters not now in the plottable font. These data are also of importance in regard to 'normalizing' the vectors representing different occurrences of a given radical of the same position type in different characters. This is a problem in Kuno's original data, where vector specifications of the same radical in the same location are often found to differ considerably from character to character; this is no doubt due to the fact that the task of drawing the characters on the matrices was shared by a number of different people, each of whom had his or her own sense of proportions. No attempt was made, it seems, to standardize the representations of common radicals or other frequently occurring elements, and thus in the data we find, for example, that radical 18, 刀/刂, which occurs most commonly on the right-hand side of characters ( ), has received 34 different codings in the vector sets of the 66 characters in the Standard Telegraphic Code in which it appears in this form.

'Normalizing' the vector sets representing radicals or other frequently occurring graphic elements in the font is not too difficult an undertaking, given the character-composition capabilities which we have developed; in effect, normalizing means throwing away an old vector specification for a character and replacing it with a new set of data which indicate not just the beginning and end points of each line or stroke in the character, but rather the components or parts of which the character is composed as well as the particular window structure arrangement according to which its parts can be assembled to form the whole character.

Having respecified the character-vector data in this form, it then becomes possible to go in the other direction, that is, to reduce the entire font to its primitives. Then, by precision digitizing of a new set of primitives, the entire font could be replaced by something a little more esthetically desirable than Kuno-style characters, which, although quite legible, are not entirely pleasing to the eye.

Following is a sample taken from initial tests of our routines for assembling characters from components utilizing the window structure approach.

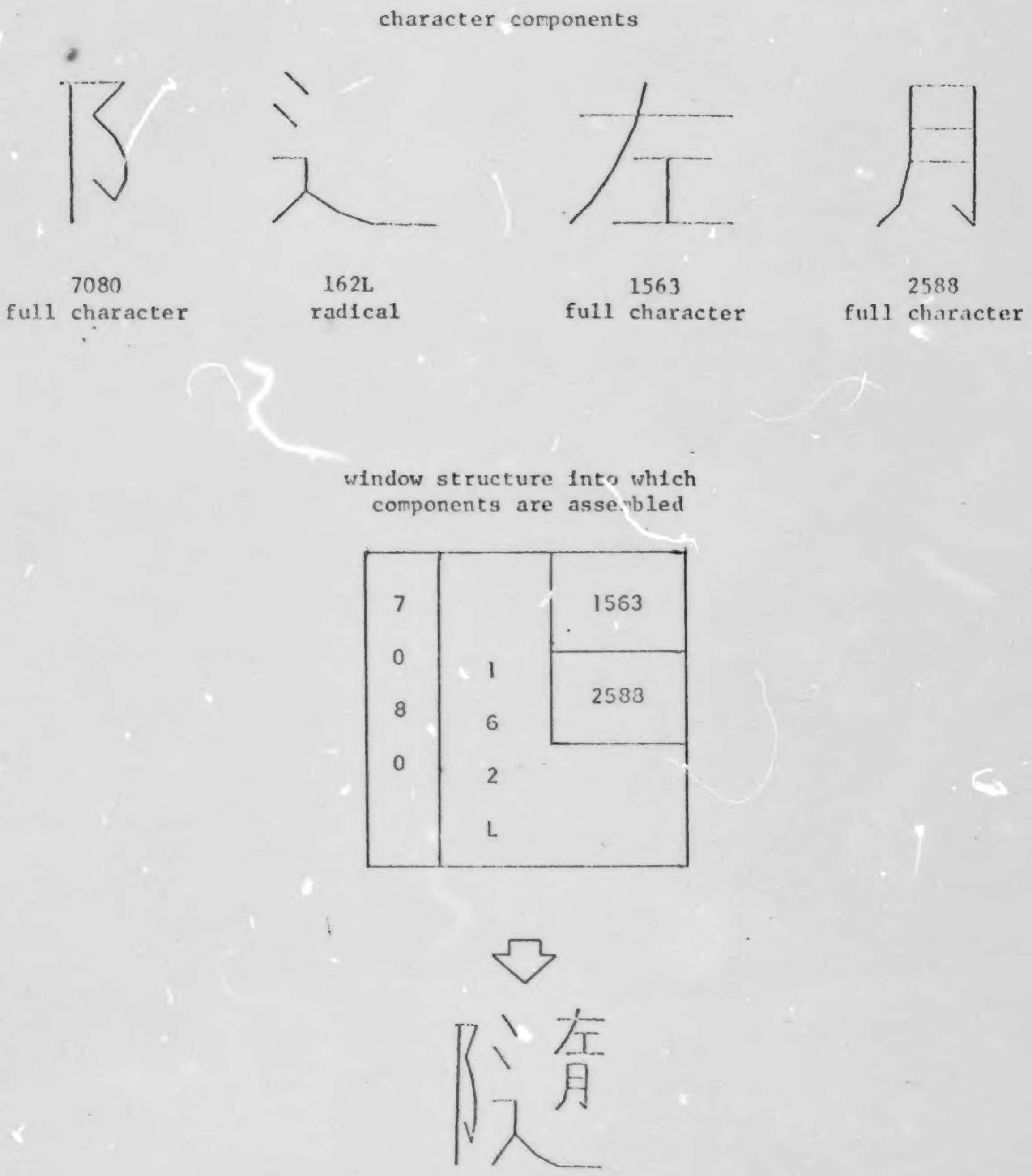
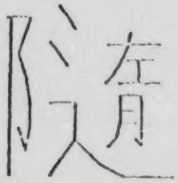

character components
7080
full character
162L
radical
1563
full character
2588
full character
window structure into which
components are assembled
7080
162L
1563
2588

Figure 3-8. Assembling Vectors for Plotting Chinese Characters

It should be pointed out that in the example in Figure 3-8 , the dimensions of the 'window' containing elements 1563 and 2588 were not correctly specified, and that the assembled plotted character is not as esthetically desirable as it could be. By slightly altering the dimensions of this window, we reassemble to obtain



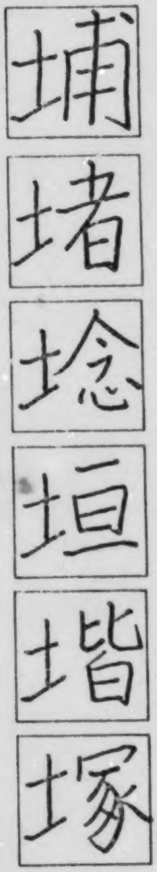
The actual coded data from which the character in the example was assembled:

```
7151      +
7151       (A,A/F,P = *7080)
7151       (F,A/P,P = *162L (F,D/P,P =
7151       (F,J/P,P = *1563)(F,D/P,J = *2588)
7151       ))
7151      -
```

Kuno's original vector set for the same character is given as

```
7151  170  13

      AOEO  EOCK  CKEH  EHEG  EGDE  DECD  CDBE  BABO
      FNHL  EJGL  EGGG  GGGC  GCEA  GCJA  JAOA  HNON
      KPJL  JLHJ  JLNL  LLLJ  NJJJ  JHJA  JHNH  NHNA
      NALB  NDJD  JFNF
```



```
1033          +
1033  032L         (A,A/F,P = *032L)
1033               (F,A/P,P = *3940)
1033          -

1035          +
1035  032L         (A,A/F,P = *032L)
1035               (F,A/P,P = *5074)
1035          -

1036          +
1036  032L         (A,A/F,P = *032L)
1036               (F,A/P,P = *1819)
1036          -

0997          +
0997  032L         (A,A/F,P = *032L)
0997               (F,A/P,P = *8017)
0997          -

1038          +
1038  032L         (A,A/F,P = *032L)
1038               (F,A/P,P = *4105)
1038          -

1046          +
1046  032L         (A,A/F,P = *032L)
1046               (F,A/P,P = *0386)
1046          -
```

Figure 3- 9. Coding for Characters to be
Added to Plottable Font

Because of the significant contribution that the ability to plot Chinese characters would make to the Project's linguistic efforts, it is with disappointment that we must report that the Project is still without finished plotting routines (due to non-completion of the programming contracted to Dynamic Graphics). Working with text concordances represented only in telecode form without Chinese characters is a very time-consuming and arduous task for any linguist without an intimate knowledge of the Standard Telegraphic Code, and even though we are now able to automatically romanize such text files, the romanization is only a poor and often very ambiguous substitute for the characters themselves. The Project's work with Chinese concordances and other text-based research will not be able to progress at the rate earlier projected and desired until enough time and funds become available to ensure the completion of the character-plotting package and to create a prototype Chinese text-editing terminal.

8. Telecode Substitution and Text Normalization

Routines to perform telecode substitutions and to normalize Chinese text have been designed, coded, debugged and tested, and have been or are currently being incorporated into the front end of the QUINCE system, where they contribute significantly to the system's ability to regularize or normalize the coded Chinese text upon which it works by manipulating this text into well-delimited and clearly identified units for subsequent processing by the rest of the system.

The telecode substitution package performs m-to-n mapping of telecodes, where m is any number of telecodes from 1 to 14 and n is any number of telecodes from 0 to 14. Below are three examples of substitutions carried out by the telecode substitution routines:

圕 → 图 书 館
0959 → 0956 2579 5290

999P 0001 → 0000 [= deletion] (999P, a text-formatting telecode used in text coding, means 'following telecode is page number'; page numbers and their indicators are deleted from text before further processing.)

伽 瑪 → γ
8026 3854 → 994C [Greek gamma]

The telecode substitution routines are also able to distinguish between and treat differently text strings (a text string is a string of one or more telecodes) surrounded by parentheses or square brackets which contain numerals and parenthesized or bracketed text strings which do not contain numerals, i.e., ordinary parenthetical expressions in running text. For example, in the string

一	种	波	長	較	長	之	光	(	燐	光	)
0001	4429	3134	7022	6525	7022	0037	0342	9988	3592	0342	9989
1	2	3	4	5	6	7	8	9	10	11	12

'a kind of light with relatively long wavelength (phosphorescence)'

the telecode substitution routines identify the content of the parentheses as an ordinary parenthetical expression and segment the text string into four parse units:

string	parse unit
1-8	1
9	2
10-11	3
12	4

whereas in the text string

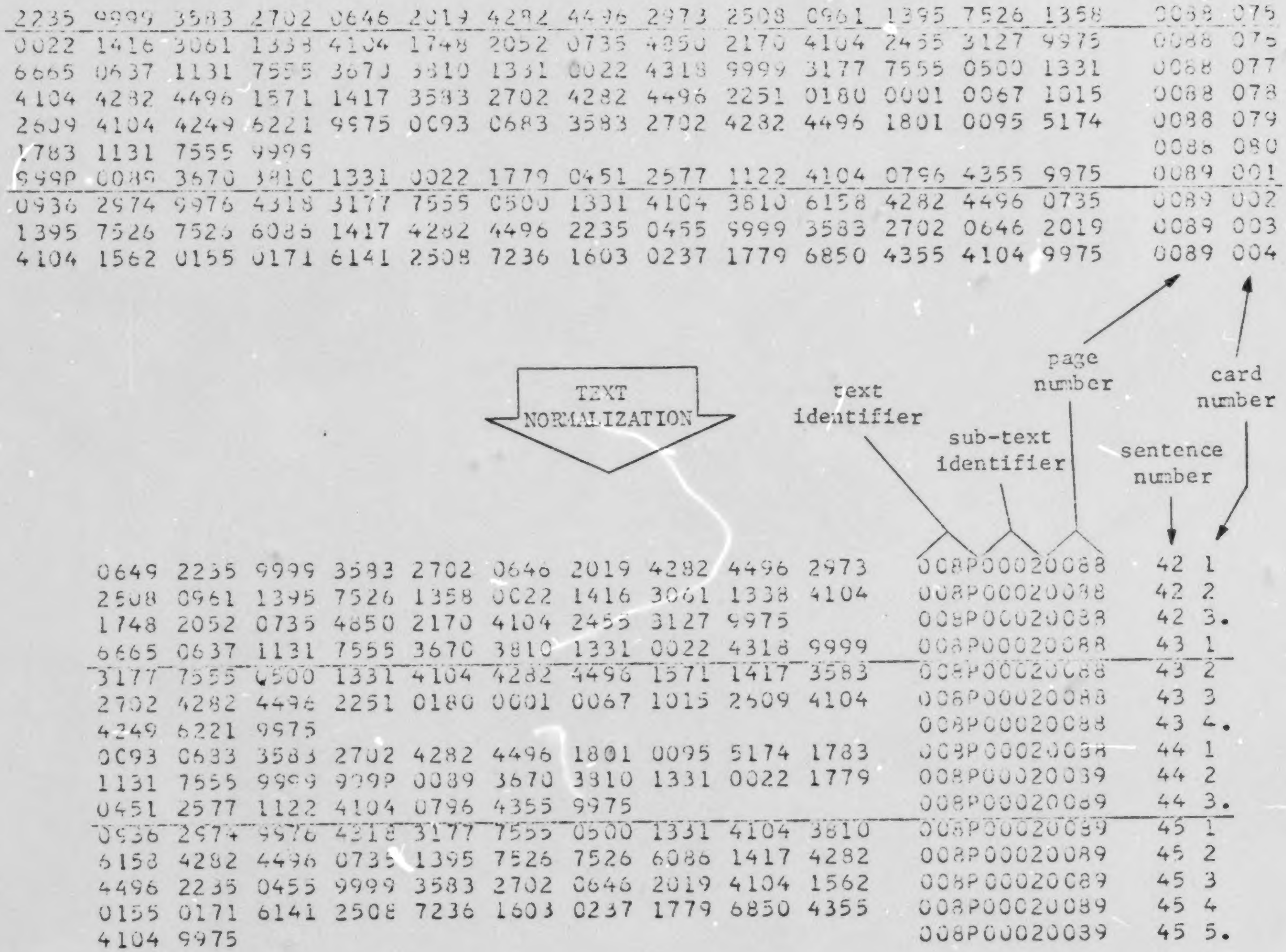
如	式	(	3	2	)	所	示	,	
1172	1709	9988	995C	995B	9989	2076	4355	9976	. . .

'as is indicated by equation (32), . . .'

a different treatment of the parenthetical expression '(32)' is required, as the expression plays a definite syntactic role in the parse unit in which it is embedded, and thus it must remain on hand for subsequent parsing. In cases of this type, the telecode substitution routines replace the original parentheses telecodes (9988, 9989) with two other special telecodes (9950, 995P), thus leaving the string '(32)' in place but identified as a special case which can be handled by a special sub-grammar during later parsing.

Our text-normalizing routines serve to break the original coded Chinese text up into graphic sentences based on the sentential-level punctuation which the text contains, and to assign to each sentence a number indicating its text of origin and another number indicating its sequential position in that text (see Figure 3-10); the text-segmenting routines then further divide these sentences into parse units by locating sub-sentence-level punctuation (colons, semicolons, commas, etc.) and other intrasentential textual features. It is these parse units which later come to constitute the basic units of analysis during dictionary lookup and parsing.

The ability of the text-normalizing package to assign unique identification labels to sentences paves the way for creating a 'sentence-feature-oriented' text corpus, wherein each sentence, in addition to its identifier, has attached to it a description of its syntactic, semantic, lexical, or other features or attributes of linguistic interest; sentences possessing similar feature descriptions may then be retrieved from the corpus as desired, which will decidedly benefit detailed linguistic study of particular problems at the sentence level. The sentence-feature-oriented text corpus might also serve as a repository of some of the kinds of information about sentences which could constitute basic data in the larger scope of discourse analysis. During the present contract period, we have given some thought to establishing a sentence-feature-oriented text corpus should funds to do so become available.



```
2235 9999 3583 2702 0646 2019 4282 4496 2973 2508 0961 1395 7526 1358    0088 075
0022 1416 3061 1338 4104 1748 2052 0735 4850 2170 4104 2455 3127 9975    0088 076
6665 0637 1131 7555 3670 3810 1331 0022 4318 9999 3177 7555 0500 1331    0088 077
4104 4282 4496 1571 1417 3583 2702 4282 4496 2251 0180 0001 0067 1015    0088 078
2609 4104 4249 6221 9975 0093 0683 3583 2702 4282 4496 1801 0095 5174    0088 079
1783 1131 7555 9999                                                      0088 080
999P 0089 3670 3810 1331 0022 1779 0451 2577 1122 4104 0796 4355 9975    0089 001
0936 2974 9976 4318 3177 7555 0500 1331 4104 3810 6158 4282 4496 0735    0089 002
1395 7526 7526 6086 1417 4282 4496 2235 0455 9999 3583 2702 0646 2019    0089 003
4104 1562 0155 0171 6141 2508 7236 1603 0237 1779 6850 4355 4104 9975    0089 004
```

```
0649 2235 9999 3583 2702 0646 2019 4282 4496 2973   008P00020088   42 1
2508 0961 1395 7526 1358 0022 1416 3061 1338 4104   008P00020088   42 2
1748 2052 0735 4850 2170 4104 2455 3127 9975        008P00020088   42 3.
6665 0637 1131 7555 3670 3810 1331 0022 4318 9999   008P00020088   43 1
3177 7555 0500 1331 4104 4282 4496 1571 1417 3583   008P00020088   43 2
2702 4282 4496 2251 0180 0001 0067 1015 2609 4104   008P00020088   43 3
4249 6221 9975                                      008P00020088   43 4.
0093 0683 3583 2702 4282 4496 1801 0095 5174 1783   008P00020088   44 1
1131 7555 9999 999P 0089 3670 3810 1331 0022 1779   008P00020089   44 2
0451 2577 1122 4104 0796 4355 9975                  008P00020089   44 3.
0936 2974 9976 4318 3177 7555 0500 1331 4104 3810   008P00020089   45 1
6158 4282 4496 0735 1395 7526 7526 6086 1417 4282   008P00020089   45 2
4496 2235 0455 9999 3583 2702 0646 2019 4104 1562   008P00020089   45 3
0155 0171 6141 2508 7236 1603 0237 1779 6850 4355   008P00020089   45 4
4104 9975                                           008P00020089   45 5.
```

Figure 3-10. Normalized Text Sample

References

Chang Shi-kuo: "An Interactive System for Chinese Character Generation and Retrieval," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-3, No. 3, May, 1973.

Hideyuki Hayashi, Sheila Duncan, and Susumo Kuno: "Graphical Input/Output of Nonstandard Characters," Communications of the ACM, Vol. 11, No. 9, September, 1968.

Leban, Carl: "Graphemic Synthesis," Proceedings of the CETA Workshop on Chinese Language and Chinese Research Materials, CETA-72-01, July, 1972.

Lin Shu, et al.: "Zhong-wen Ji-suan-ji zhi Zi-gen Ji-xiao-ji Jia-ma ji Mei-hua Yan-jiu" (Coding and 'Beautification' of Minimum Set of Character Components for Chinese Computer [input and output]), Ke-xue Fa-zhan (Scientific Development), Vol. 2, No. 11, November, 1974.

Walker, Gordon L., Susumo Kuno, Barbara N. Smith, and Roland B. Holt: "Chinese Mathematical Text Analysis," IEEE Transactions on Engineering Writing and Speech, Vol. EWS-11, No. 2, August, 1968.

Zhuang Yao-min: "Dian-nao-hua de Gao-su Ying-pai-ji" (Computerized High-Speed [Chinese] Phototypesetter), Jin-ri Shi-jie (Today's World), No. 12, December, 1974.

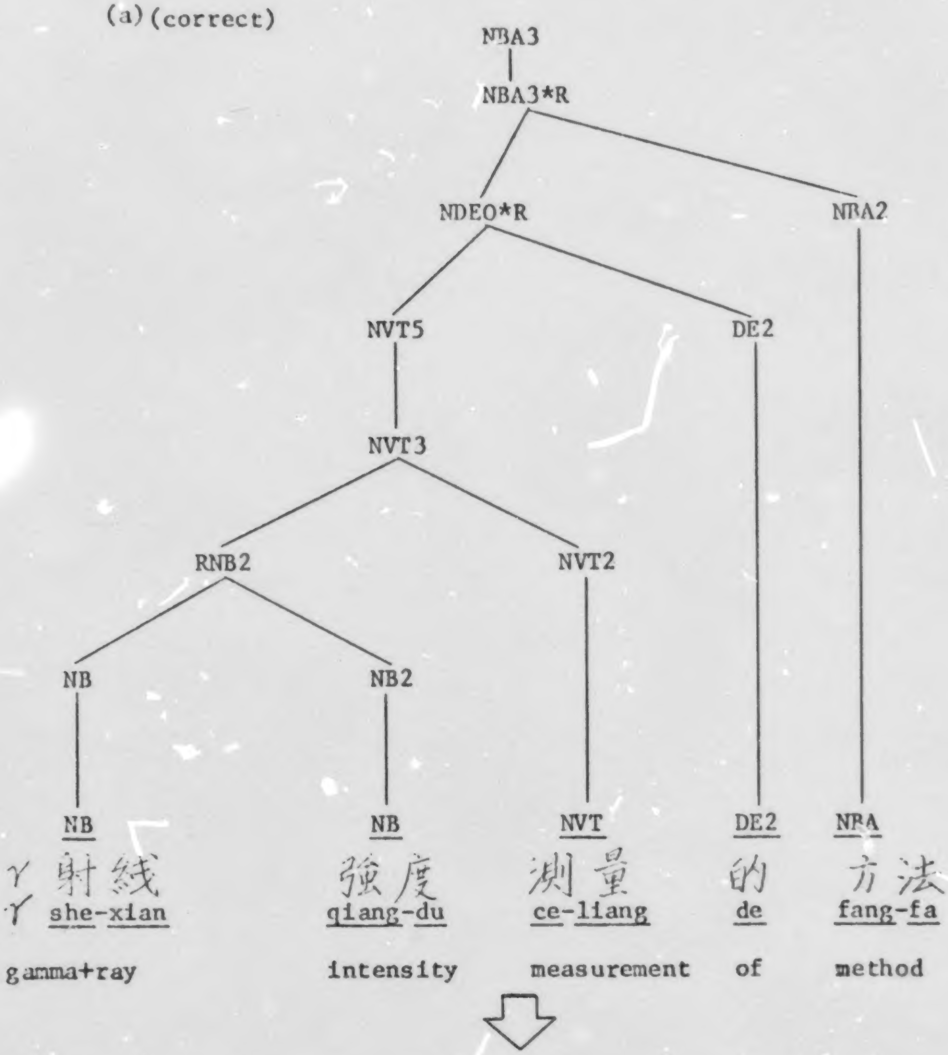
IV. STUDIES IN INTERLINGUAL SYNTAX AND SEMANTICS

1. Semantic Research

During the recent contract period, work has progressed steadily on the semantic categorization of nouns and verbs. We have found two major discovery procedures to be useful in this area of research. The "static" procedure consists in examining particular lexical entries and consists in assigning features on the basis of general "knowledge of the world" and also the *probable* importance of such features to unambiguous parsing. The "dynamic" procedure, on the other hand, involves searching available texts for strings which are *certain* to parse ambiguously unless a given lexical item is assigned a particular semantic feature. Of the two procedures, the second is the more tedious but yields by far the more valuable results, since only it can provide the absolute measure of how much "knowledge of the world" our parser must incorporate. To put it another way, a linguist examining lexical items out of context cannot possibly predict all the ways in which such items will combine in an actual text. Yet it is just these unpredicted combinations that will prove disastrous if parsing of the text is attempted. Our "dynamic" procedure thus provides an important check on the linguist's intuitions concerning the semantic capabilities of the parser. The following strings illustrating problems in semantic disambiguation are typical of those encountered in textual analysis. They will be shown as parsed before and after the reassignment of semantic features:

Figure 1: before semantic disambiguation

(a)(correct)

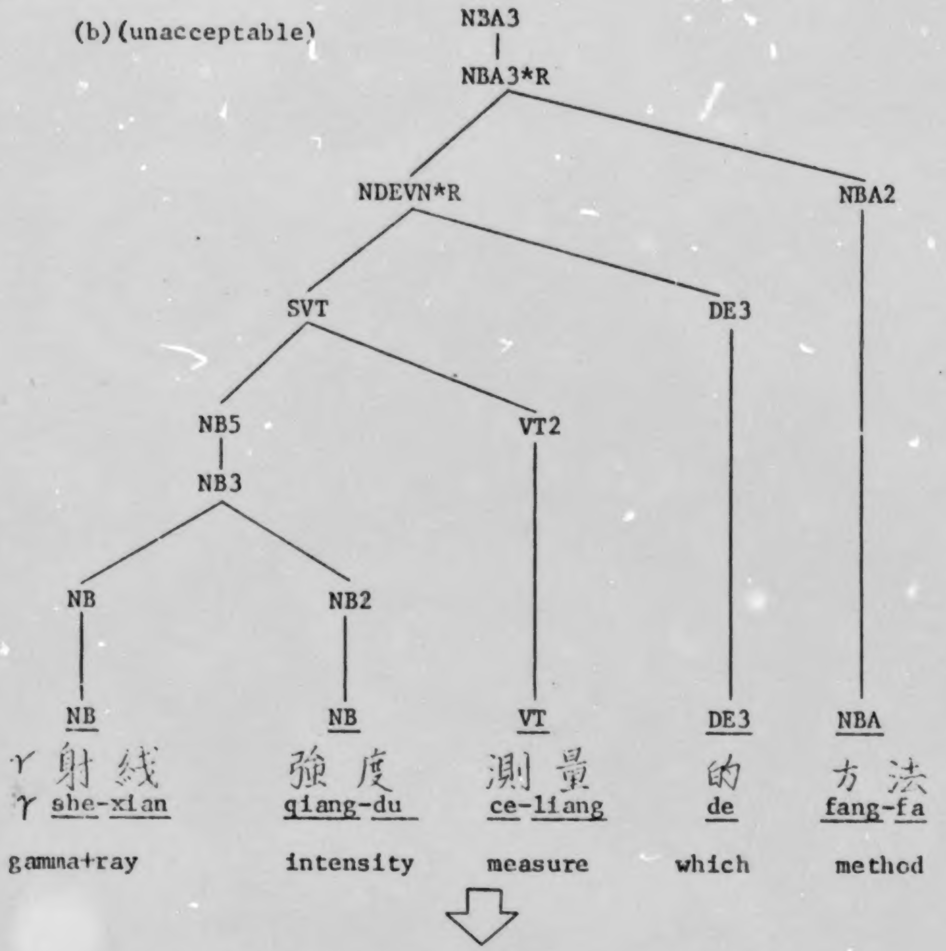

NBA3
NBA3*R
NDEO*R
NBA2
NVT5
DE2
NVT3
RNB2
NVT2
NB
NB2
NB
NB
NVT
DE2
NBA
γ 射 線
γ she-xian
gamma+ray
強 度
qiang-du
intensity
測 量
ce-liang
measurement
的
de
of
方 法
fang-fa
method

'method of gamma-ray intensity measurement'

Interlingual
- (a) NDEO*R permutes NVT5 'gamma-ray intensity measurement' and DE2 'of'.
- (b) NBA3*R permutes NBA2 'method' and NDEO*R 'of gamma-ray intensity measurement'.

Figure 2: before semantic disambiguation

(b)(unacceptable)


NBA3
NBA3*R
NDEVN*R
NBA2
SVT
DE3
NB5
VT2
NB3
NB
NB2
NB
NB
VT
DE3
NBA
γ射線
γ she-xian
gamma+ray
強度
qiang-du
intensity
測量
ce-liang
measure
的
de
which
方法
fang-fa
method

'method which measure gamma-ray intensity'

Interlingual

(a) NDEVN*R permutes SVT 'gamma-ray intensity measure' and DE3 'which'

(b) NBA3*R permutes NDEVN*R 'which gamma-ray intensity measure' and NBA2 'method'.

Semantic disambiguation: Since VT *ce-liang* 'measure' can take only [+human] subjects, its concatenation with [+abstract] NB5 γ *she-xian qiang-du* 'gamma-ray intensity' is semantically contradictory. In order to reflect this information, we must change the grammar code of VT *ce-liang* 'measure' to VTH, where H stands for [+human] subject. If we now attempt to parse the original surface string again, unacceptable structure (b) will fail to be generated because of parse abort (indicated by the symbol Ø) where there is no rule to conjoin [+human subject] VTH *ce-liang* 'measure' with the supposed [+abstract] subject NB5 γ *she-xian qiang-du* 'gamma-ray intensity' (see aborted structure b'). Only the correct structure, which now reads as in (a'), will survive:

Figure 3: after semantic disambiguation

(b')(aborted parse)

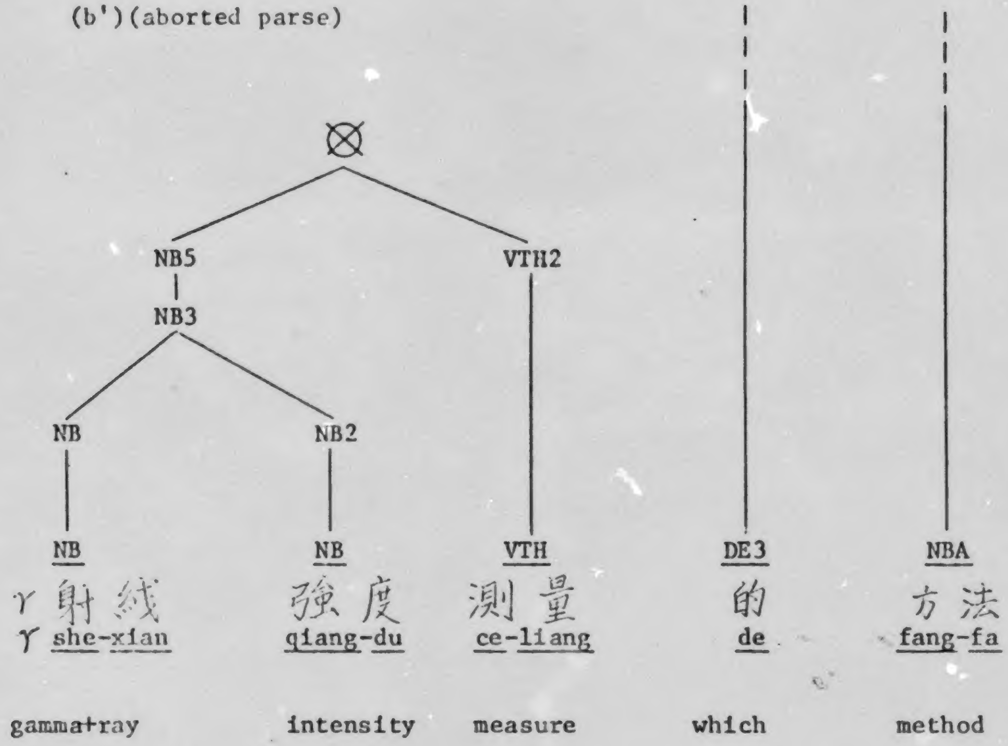
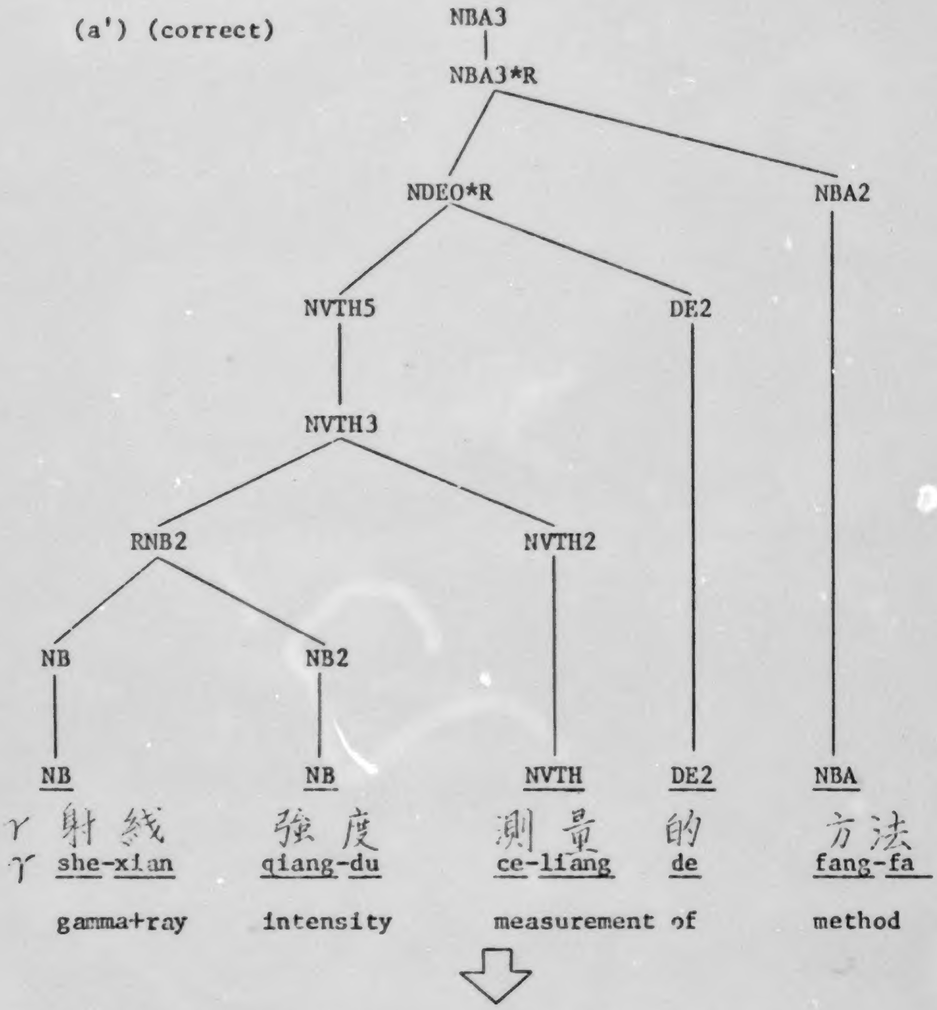

NB5
VTH2
NB3
NB
NB2
NB
NB
VTH
DE3
NBA
γ 射 綫
γ she-xian
強 度
qiang-du
測 量
ce-liang
的
de
方 法
fang-fa
gamma+ray
intensity
measure
which
method

Figure 4: after semantic disambiguation

(a') (correct)

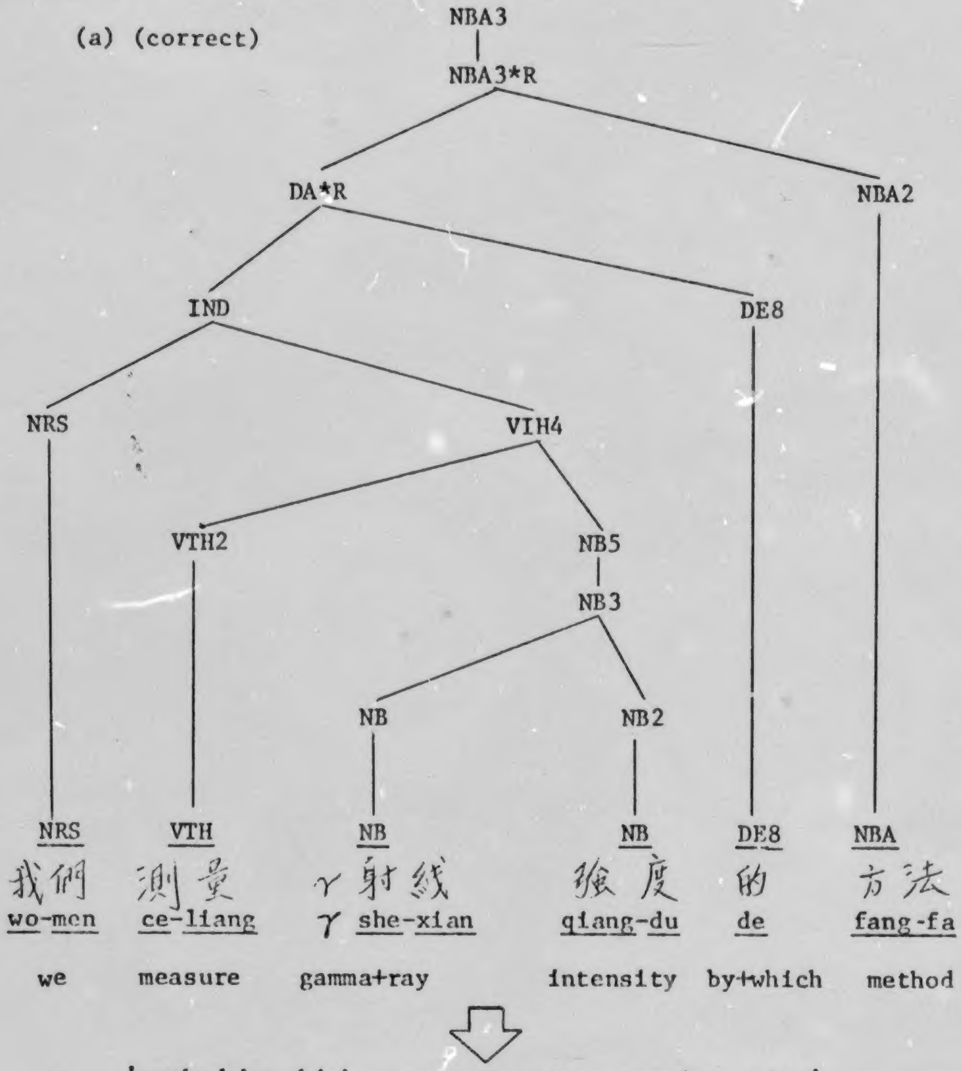

NBA3
NBA3*R
NDEO*R
NBA2
NVTH5
DE2
NVTH3
RNB2
NVTH2
NB
NB2
NB
NB
NVTH
DE2
NBA
γ 射 線
γ she-xian
強 度
qiang-du
測 量
ce-liang
的
de
方 法
fang-fa
gamma+ray
intensity
measurement
of
method

'method of gamma-ray intensity measurement'

Interlingual operations: as in Figure 1.

Figure 5: before semantic disambiguation

(a) (correct)


NBA3
NBA3*R
DA*R
NBA2
IND
DE8
NRS
VIH4
VTH2
NB5
NB3
NB
NB2
NRS
VTH
NB
NB
DE8
NBA
我們
測量
γ射线
強度
的
方法
wo-men
ce-liang
γ she-xian
qiang-du
de
fang-fa
we
measure
gamma+ray
intensity
by+which
method

'method by which we measure gamma-ray intensity'

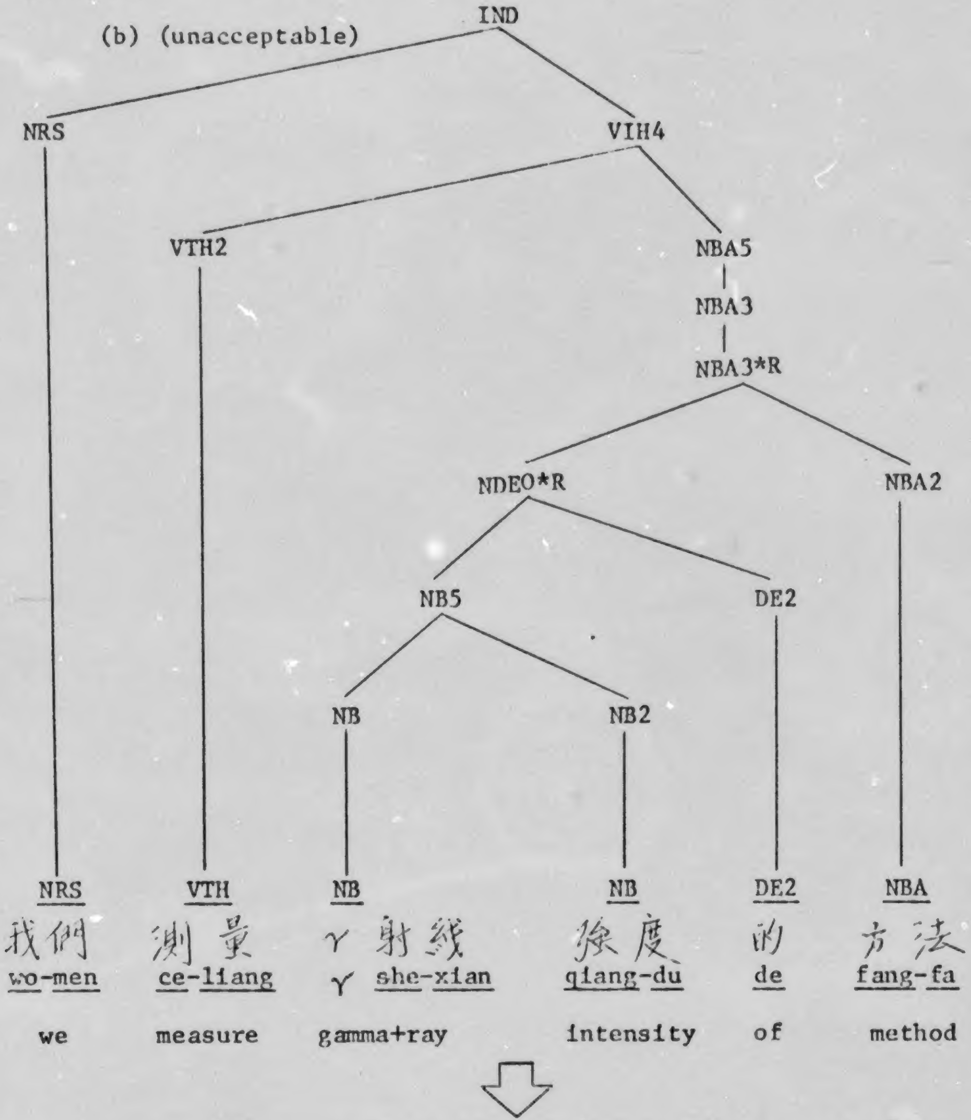
Interlingual

(a) DA*R permutes IND 'we measure gamma-ray intensity' and DE8 'by which'.

(b) NBA3*R permutes DA*R 'by which we measure gamma-ray intensity' and NBA2 'method'.

Figure 6: before semantic disambiguation

(b) (unacceptable)


IND
NRS
VIH4
VTH2
NBA5
NBA3
NBA3*R
NDEO*R
NBA2
NB5
DE2
NB
NB2
NRS
VTH
NB
NB
DE2
NBA
我們 wo-men we
測量 ce-liang measure
γ射綫 γ she-xian gamma+ray
強度 qiang-du intensity
的 de of
方法 fang-fa method

*'we measure the method of gamma-ray intensity'

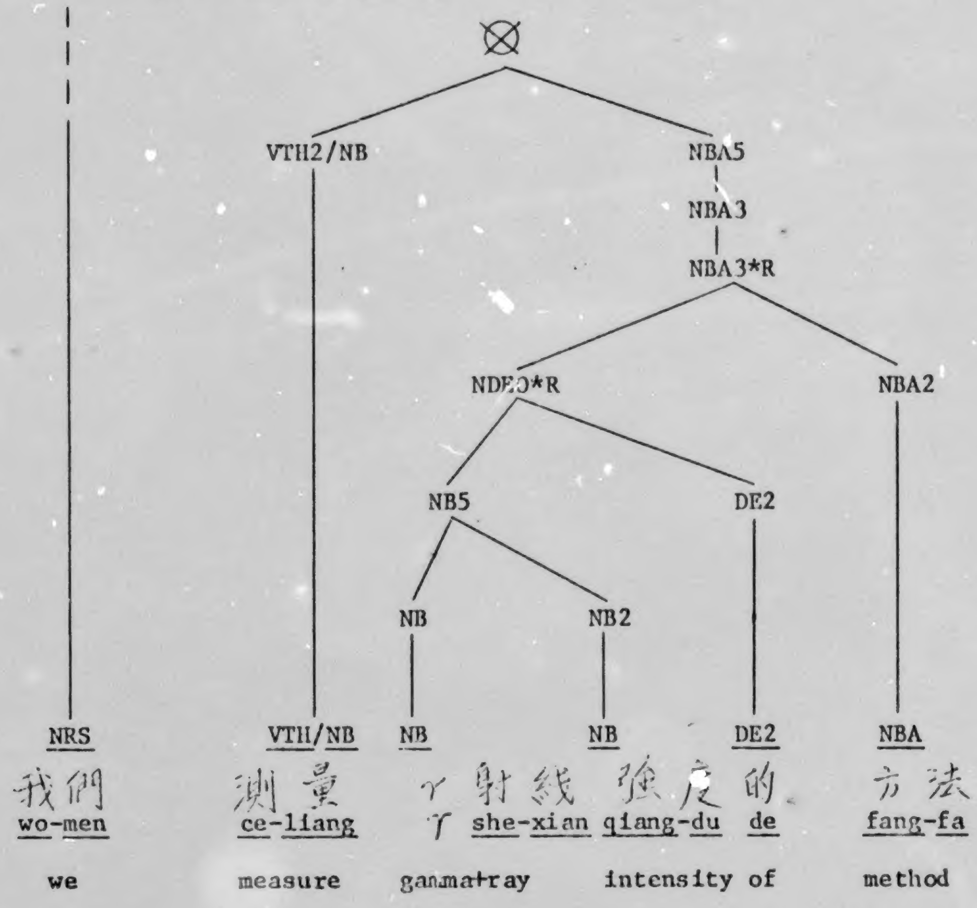
Interlingual

(a) NDEO*R permutes NB5 'gamma-ray intensity' and DE2 'of'.
(b) NBA3*R permutes NDEO*R 'of gamma-ray intensity' and NBA2 'method'.

Semantic disambiguation: We have already revised VT *ce-liang* 'measure' to VTH in order to reflect restrictions on possible subjects (cf. example (1) of this set). In the present example, however, it is the restrictions on possible *objects* of *ce-liang* 'measure' that are most relevant to semantic disambiguation: *ce-liang* 'measure' can take only [+abstract] objects which are also inherent properties of physical entities. Nouns with the grammar code NBA (e.g. 重要性 *zhong-yao-xing* 'importance', 原故 *yuan-gu* 'reason', 觀念 *guan-nian* 'concept', 问題 *wen-ti* 'problem, question', as well as 方法 *fang-fa* 'method') are abstract, but cannot be inherent properties of physical entities and thus do not fit the requirements just mentioned. In order to reflect this information, we must further revise the grammar code of *ce-liang* 'measure' to VTH/NB, where /NB stands for [+abstract object] other than NBA. If we now attempt to parse the original surface string again, unacceptable structure (b) will fail to be generated because of parse abort (indicated by ⊠) where there is no rule to conjoin VTH/NB *ce-liang* 'measure' with the supposed object NBA *fang-fa* 'method' (see aborted structure b'). Only the correct structure, which now reads as in (a') will survive:

Figure 7: after semantic disambiguation

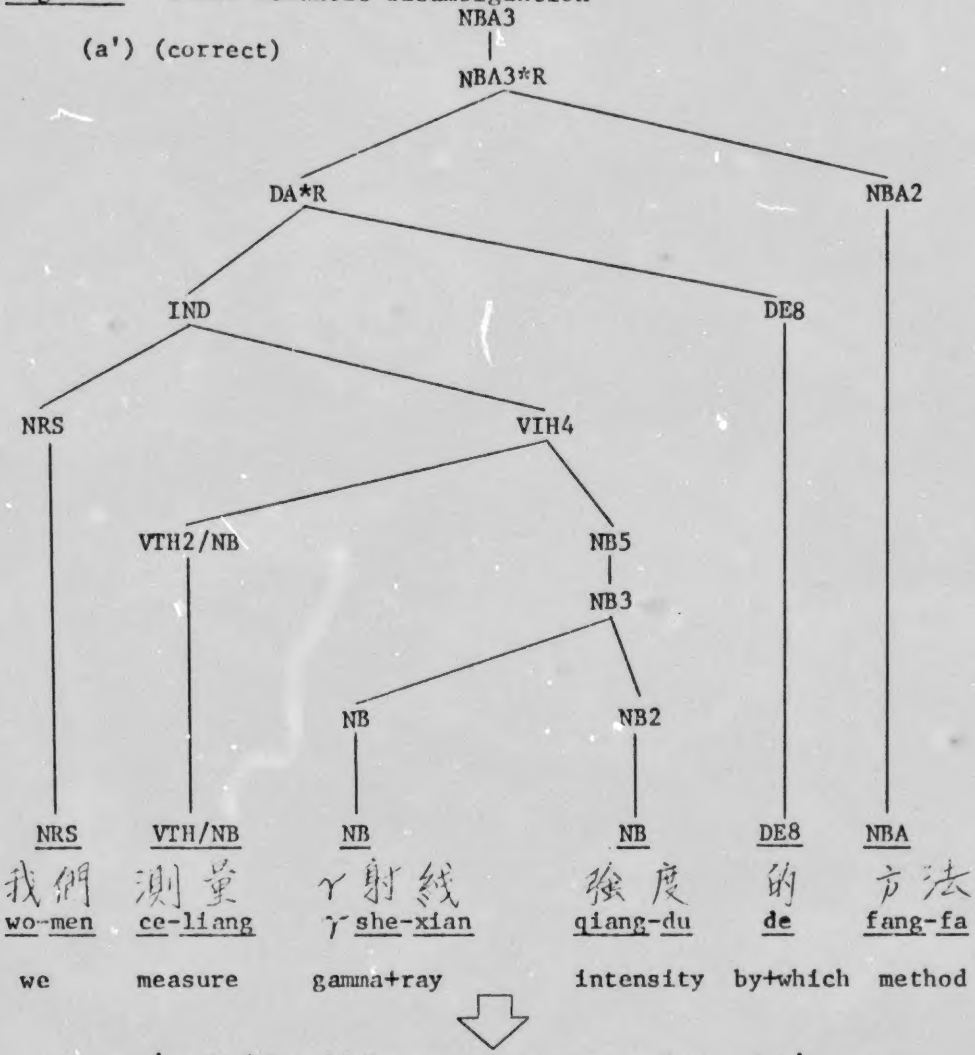
(b') aborted parse



⊗
VTH2/NB
NBA5
NBA3
NBA3*R
NDEO*R
NBA2
NB5
DE2
NB
NB2
NRS
VTH/NB
NB
NB
DE2
NBA
我們
測量
γ 射線
強度
的
方法
wo-men
ce-liang
γ
she-xian
qiang-du
de
fang-fa
we
measure
gamma-ray
intensity
of
method

Figure 8: after semantic disambiguation

(a') (correct)


NBA3
NBA3*R
DA*R
NBA2
IND
DE8
NRS
VIH4
VTH2/NB
NB5
NB3
NB
NB2
NRS
VTH/NB
NB
NB
DE8
NBA
我們
wo-men
測量
ce-liang
γ射线
γ she-xian
強度
qiang-du
的
de
方法
fang-fa
we
measure
gamma+ray
intensity
by+which
method

'method by which measure gamma-ray intensity'

Interlingual operations: as in Figure 5.

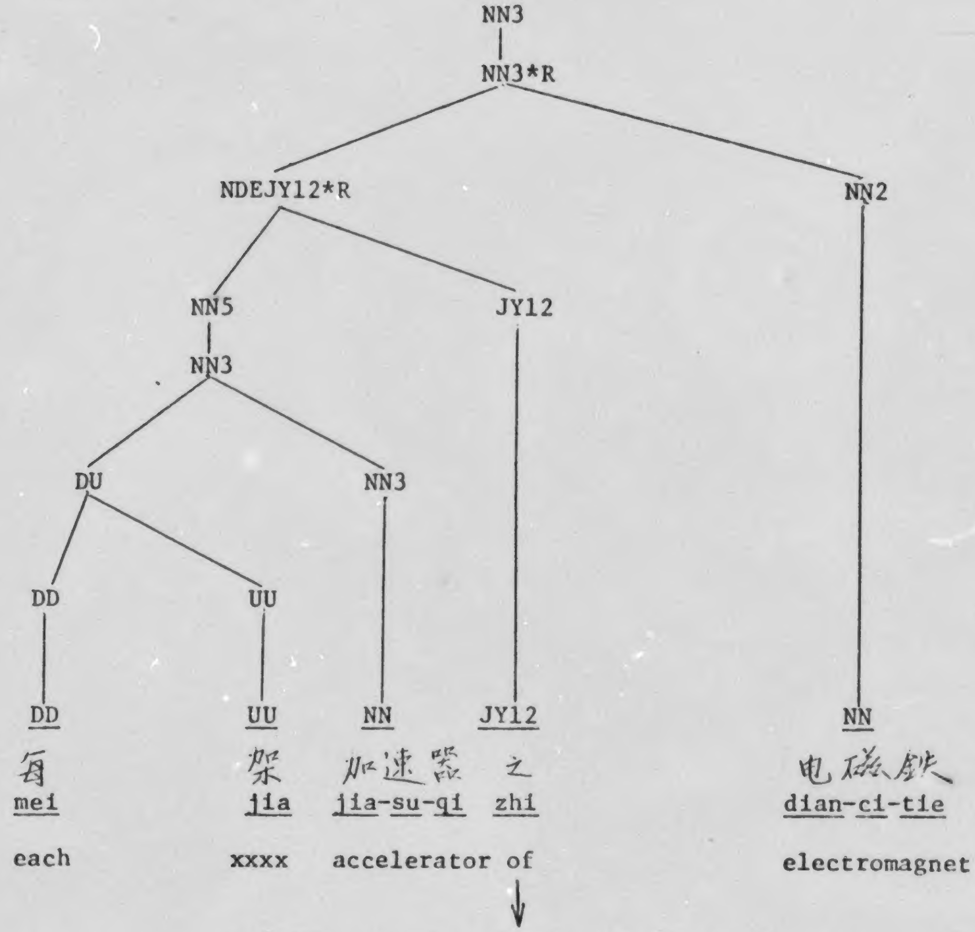
Noun classifiers are another area in which detailed semantic research is being conducted. In Chinese, all quantifiable nouns must normally be preceded by an auxiliary noun, or "classifier", along with the appropriate numerical expression. In other words, the mass-count noun distinction which holds for many 'Western' languages does not exist in Chinese in syntactic terms. Nevertheless, for purposes of string disambiguation it is still very useful to know in semantic terms which classifiers are appropriate for which nouns, and in fact a mass-count categorization based on semantic features is fairly easy to set up. Typically, any one mass noun in Chinese can take a variety of classifiers depending on its semantic characteristics; thus you-kuang 铀矿 'uranium ore' can be quantified by 'piece' (kuai 块), 'ton' (duen 噸), 'kilogram' (gong-jin 公斤), etc., just like its counter part in English. On the other hand, a Chinese count noun usually takes one specific classifier, which unlike mass-noun classifiers has no direct equivalent in English and is assigned a zero gloss at the interlingual stage. Note the following examples with the numeral san 三 'three' (classifiers are underlined twice): san jia jia-su-qi 三架加速器 'three accelerators', san zuo yuan-zi-zhong, 三座原子钟 'three atomic clocks', san zhi hang-kong-mu-jian 三只航空母舰 'three aircraft carriers',san zhan zi-wai-xian-deng 三盏紫外线灯 'three ultra-violet lamps', san gen kong-zhi-bang 三根控制棒 'three control-rods', san ge wen-ti 三个問題 'three problems', etc. Note that certain classifiers are compatible with both mass and count nouns, although the quantificatory functions are semantically quite distinct for each noun type; e.g.

Table 1

	Mass-noun classifier as in:	Count-noun classifier as in:
張 zhang	三張感光紙 san zhang gan-guang-zhi 'three sheets of photo-sensitive paper'	三張床 san zhang chuang 'three beds'
條 tiao	三條鋼條 san tiao gang-tiao 'three lengths of steel rod'	三條天綫 san tiao tian-xian 'three antennas'
塊 kuai	三塊鈾矿 san kuai you-kuang 'three pieces of uranium ore'	三塊石头 san kuai shi-tou 'three rocks'

Of course, classifiers themselves can be sub-divided according to their affinity for particular nouns. Thus, of the count-noun classifiers exemplified above, no one classifier (except possibly 个 ge) can substitute for another; likewise, a mass-noun classifier like 塊 kuai 'piece' will be semantically incompatible with certain mass nouns such as 气体 qi-ti 'gas'. This type of information can be useful for the formulation of string disambiguation procedures, as illustrated by figs. 9-10, where the scope of the count-noun classifier 架 *jia* is syntactically ambiguous. Here, it is primarily the fact that 架 *jia* is appropriate as a classifier for 加速器 *jia-su-qi* 'accelerator' rather than 电磁铁 *dian-ci-tie* 'electromagnet' that allows us to decide that the interpretation in fig. 9 is correct, whereas the one in fig. 10 is unacceptable:

Figure 9: (correct)

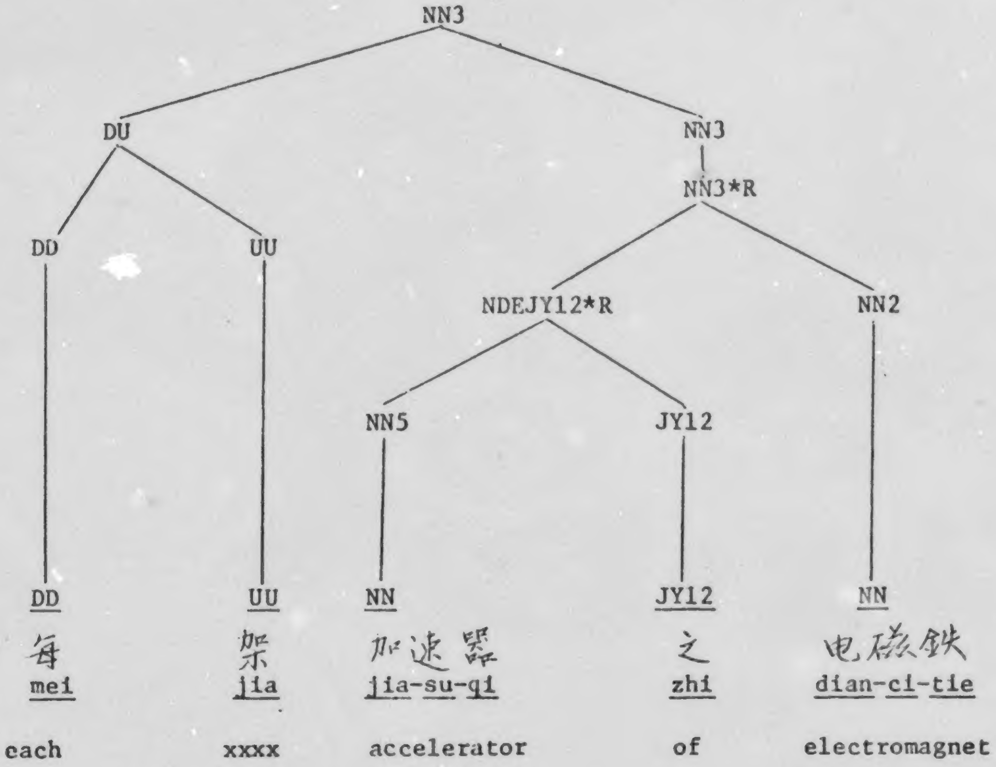

NN3
NN3*R
NDEJY12*R
NN2
NN5
JY12
NN3
DU
NN3
DD
UU
DD
UU
NN
JY12
NN
每
架
加速器
之
电磁鉄
mei
jia
jia-su-qi
zhi
dian-ci-tie
each
xxxx
accelerator
of
electromagnet

'electromagnet of each accelerator'

Interlingual operations:

(a) NDEJY12*R permutes NN5 'each accelerator' and JY12 'of'.
(b) NN3*R permutes NDEJY12*R 'of each accelerator' and NN2 'electromagnet'.

Figure 10: (unacceptable)


NN3
DU
NN3
NN3*R
DD
UU
NDEJY12*R
NN2
NN5
JY12
DD
UU
NN
JY12
NN
每
架
加速器
之
电磁铁
mei
jia
jia-su-qi
zhi
dian-ci-tie
each
xxxx
accelerator
of
electromagnet

→ *'each electromagnet of accelerator'

Interlingual operations :

(a) NDEJY12*R permutes NN5 'accelerator' and JY12 'of'.
(b) NN3*R permutes NDEJY12*R 'of accelerator' and NN2 'electromagnet'.

2. Linguistic Analysis vis-à-vis Interlingual Transfer

In the previous final technical report (Wang et al. 1974: Chapter IV), we discussed in detail current research in linguistic analysis and interlingual transfer. Such research has yielded solutions to a number of problems related to the production of readable English output, and many of these solutions have been implemented during the recent contract period. Examples will be drawn from four major syntactic areas: (A) Subordinative markers 的/之 de/zhi, (B) Locational nouns and verbs, (C) Directional verbs and complements, and (D) Function words 以/為/作 yi/wei/zuo. In each case, lexical items relevant to these areas will be presented together with a summary of their syntactic and semantic functions, and the structures in which they participate will be exemplified through a series of tree diagrams. Since it is rather unrevealing to discuss these structures in isolation, we have decided to show them in the context of an entire sentence or fairly sizeable noun or verb phrase. This means, of course, that not all parts of a tree diagram may be relevant to the demonstration of the point at hand, When in doubt, the reader should consult the index of figures at the end of each section to determine what each particular diagram is intended to illustrate.

(A) *Subordinative particles* 的 *de* and 之 *zhi* (cf. Chao 1968:285-301, Wang et. al. 1974:70-71).

Lexical types:

的 4104 *de*	DE1	's
	DE2	of
	DE3	which/that
	DE4	xxxx
	DE5	who
	DE6	that+which/those+which/what
	DE7	that+of/those+of
	DE8	by+which
	2X-AZT	xxxx
之 0037 *zhi*	JY11	's
	JY12	of
	JY3	which/that
	JY4	xxxx
	JY5	who
	JY8	by+which

The subordinative markers 的 *de* and 之 *zhi* are perhaps the most commonly occurring grammatical particles in our texts. The most frequently seen subordinative structural type is X *de*/*zhi* N, where N represents the head noun, and X represents either (a) a noun possessed by the head noun, where *de*/*zhi* are glossed 's (DE1, JY11) or 'of' (DE2, JY12), (b) a verb phrase whose subject or object is co-referent with the head noun, where *de*/*zhi* are glossed'which/that' (DE3, JY3), 'who' (DE5, JY5) or in the case of embedded adjectives, 'xxxx' (DE4, JY4), or (c) and indicative clause neither of whose subject or object is co-referent with the head noun, where *de*/*zhi* is glossed 'by which' (DE8, JY8). The X *de*/*zhi* construction may undergo ellipsis to yield X *de* where N is understood; *de* is then glossed DE6 'that which/those which/what' where X is a verb phrase, or DE7 'that of/those of' if X is a noun phrase originally taking DE2; noun phrases taking DE1 require no special gloss manipulation upon ellipsis. One other special construction type takes the form shi X de, coded SH X DE4, where X represents a verb phrase and 是 shi is glossed 'xxxx', the construction itself acting as a sort of "topicalizer" of the verb phrase in question. Where X is a noun, shi receives its more usual analysis as VC 'be' and X de is parsed as NDEP. Finally, de 2X-AZT 'xxxx' can introduce a post-verbal resultative/extent complement, in which case it is also seen written 得, also pronounced de.

The subordinative marker zhi is most frequently seen in texts where the author is adopting a classical style of writing (see Appendix 2).
Of course, many authors mix both classical and modern styles in their writing, so that in certain texts both de and zhi may appear in free variation. Consequently, in structures exemplifying the various uses of subordinative markers, it is to be understood that de and zhi can substitute for each other depending on the author's style, with the following exceptions: (1) zhi cannot replace de in its use as DE1 where NDEP functions as an elliptical noun phrase, DE4 in the shi X de construction, DE6, or DE7. (2) de cannot replace zhi in its use as JY1-4 before monosyllabic bound nouns. Note that authors adopting a more classical style will not avoid X de or shi X de constructions just in order not to use de; however, authors adopting a more modern style will avoid using X + zhi + monosyllabic bound noun, opting instead for X + de + polysyllabic free noun.

The most frequently performed interlingual operation involving de/zhi is the permutation X de/zhi N → N de/zhi X. This is accomplished in two steps: first, de/zhi and X are permuted, then the sequence de/zhi X and N. Where de/zhi are coded DE4, the first step is not performed; where de is coded DE6 or DE7 as in X de, only one permutation is necessary.

All figures except 11-14 illustrating uses of *de* and *zhi* are located before or after the present section. Note that the glosses of DE6 and DE7 are occasionally abbreviated in order to save space.

Index to Figures:

Figures 1, 4, 6, 9, 10, 12, 13, 23, 25, 27, 31, 48-57: DE2/JY12 (except JY12 with monosyllabic bound nouns).

Figures 2, 18: DE3

Figure 44: JY3

Figures 16, 19, 32, 34-36, 39: DE4

Figure 45: JY4

Figures 5, 8: DE8

Figure 11: DE1/NDEP

Figure 12: DE6/NDE6*R

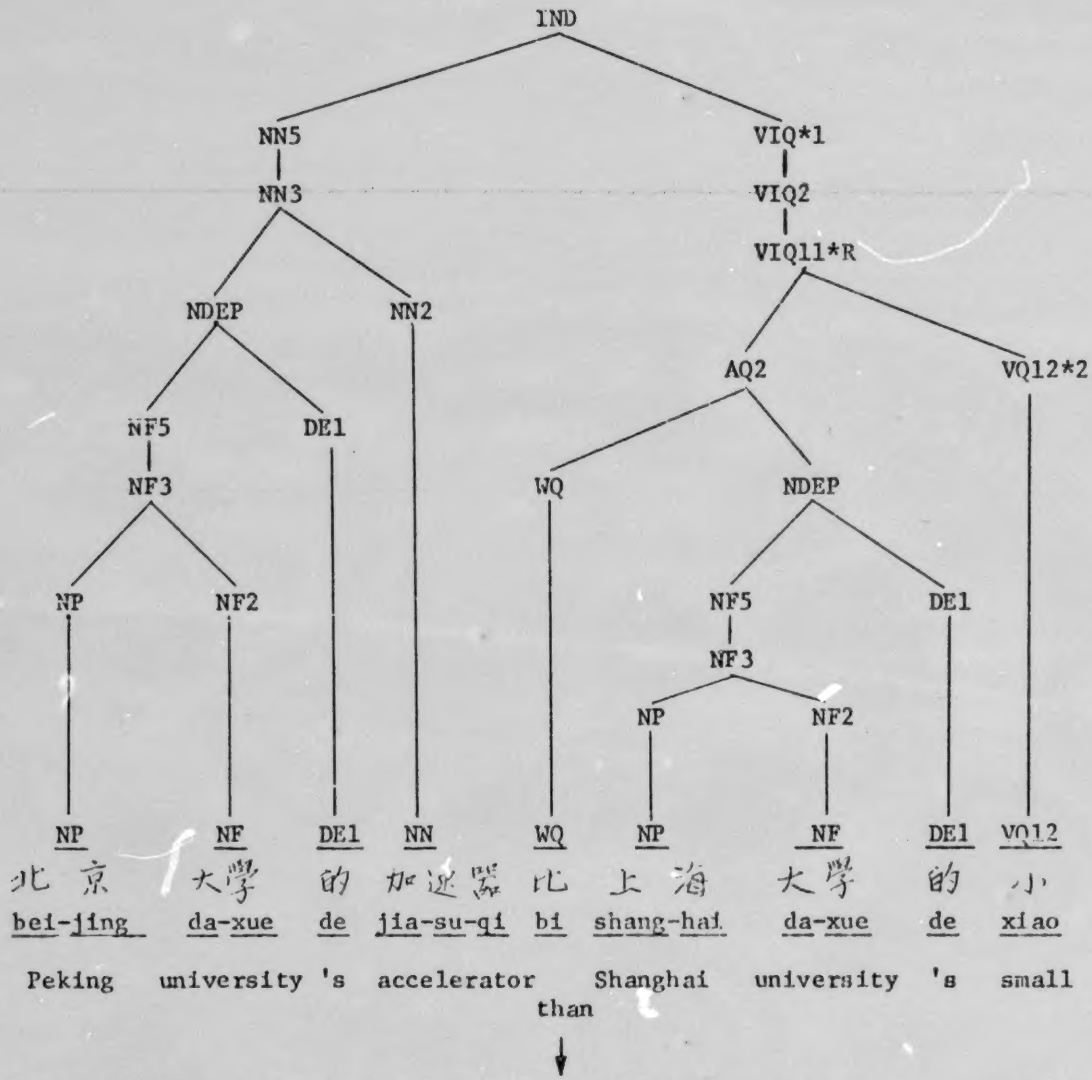
Figure 13: DE7/NDE7*R

Figure 14: 2X-AZT

Figure 24: *shi* X *de*

Figures 33, 41-43: JY12 with monosyllabic bound nouns

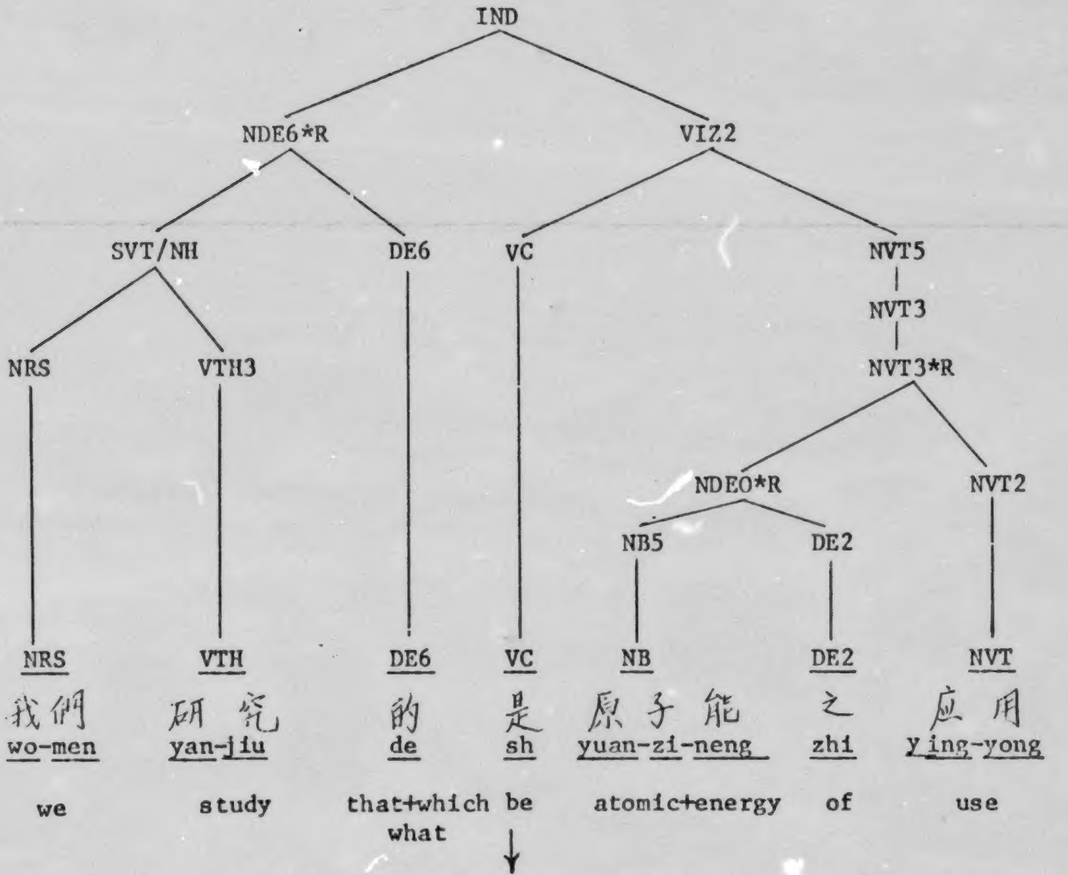
Figure 11:


IND
NN5
NN3
NDEP
NN2
NF5
NF3
NP
NF2
DE1
VIQ*1
VIQ2
VIQ11*R
AQ2
VQ12*2
WQ
NDEP
NF5
NF3
NP
NF2
DE1
NP
NF
DE1
NN
WQ
NP
NF
DE1
VQ12
北京
大學
的
加速器
比
上海
大學
的
小
bei-jing
da-xue
de
jia-su-qi
bi
shang-hai
da-xue
de
xiao
Peking
university
's
accelerator
than
Shanghai
university
's
small

'Peking University's accelerator (be) small(er) than Shanghai University's'

Interlingual operations: VIQ11*R permutes AQ2 'than Shanghai University's' and VQ12*2 '(be) small(er) than'.

Figure 12:

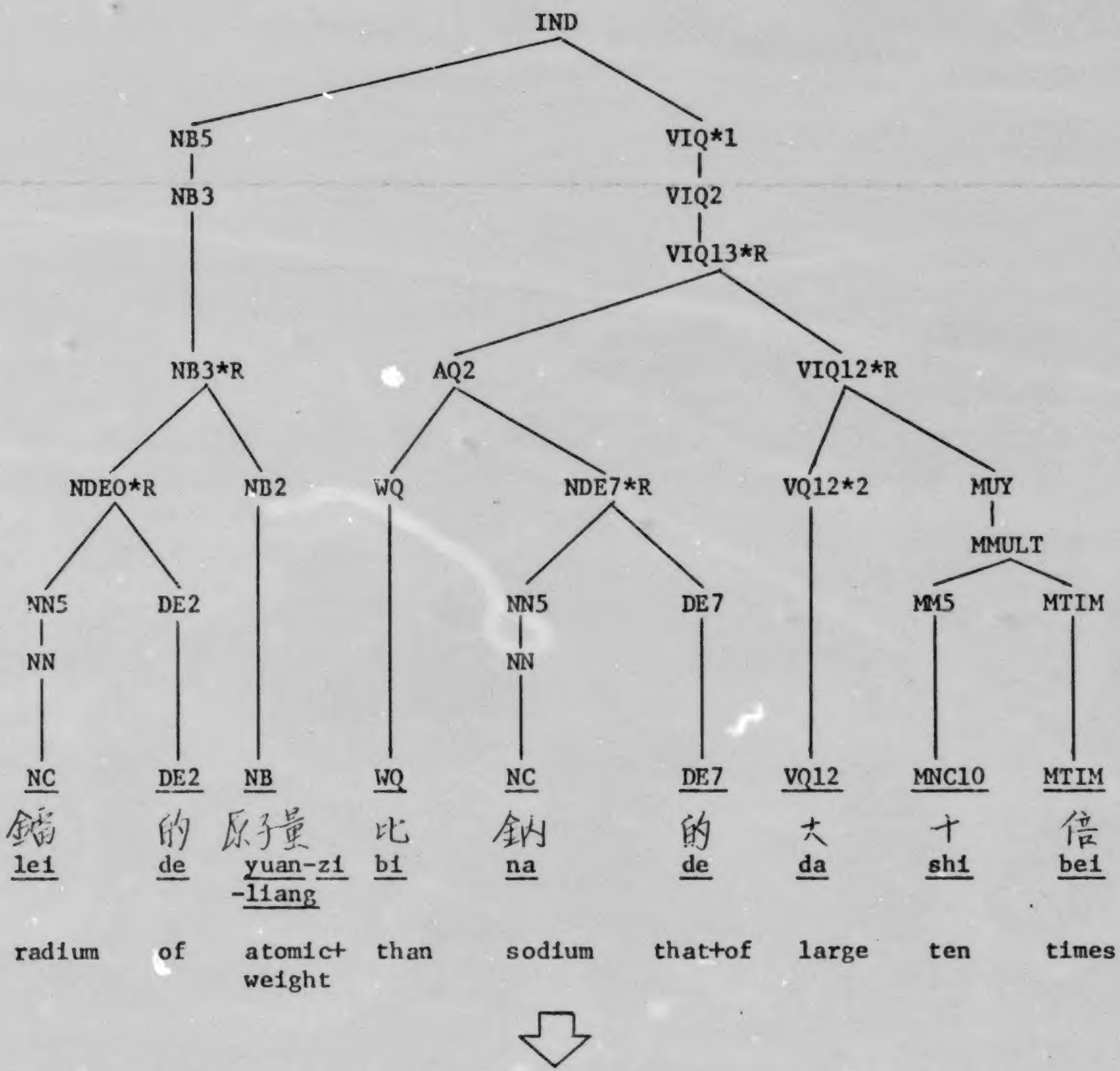

IND
NDE6*R
VIZ2
SVT/NH
DE6
VC
NVT5
NVT3
NVT3*R
NRS
VTH3
NDEO*R
NVT2
NB5
DE2
NRS
VTH
DE6
VC
NB
DE2
NVT
我們
研究
的
是
原子能
之
应用
wo-men
yan-jiu
de
sh
yuan-zi-neng
zhi
y ing-yong
we
study
that+which
what
be
atomic+energy
of
use

'that which/what we study be use of atomic energy'

Interlingual

(a) NDEO*R permutes NB5 'atomic energy' and DE2 'of'.
(b) NVT3*R permutes NDEO*R 'of atomic energy' and NVT2 'use'.
(c) NDE6*R permutes SVT/NH 'we study' and DE6 'that which/ what'.

Figure 13:

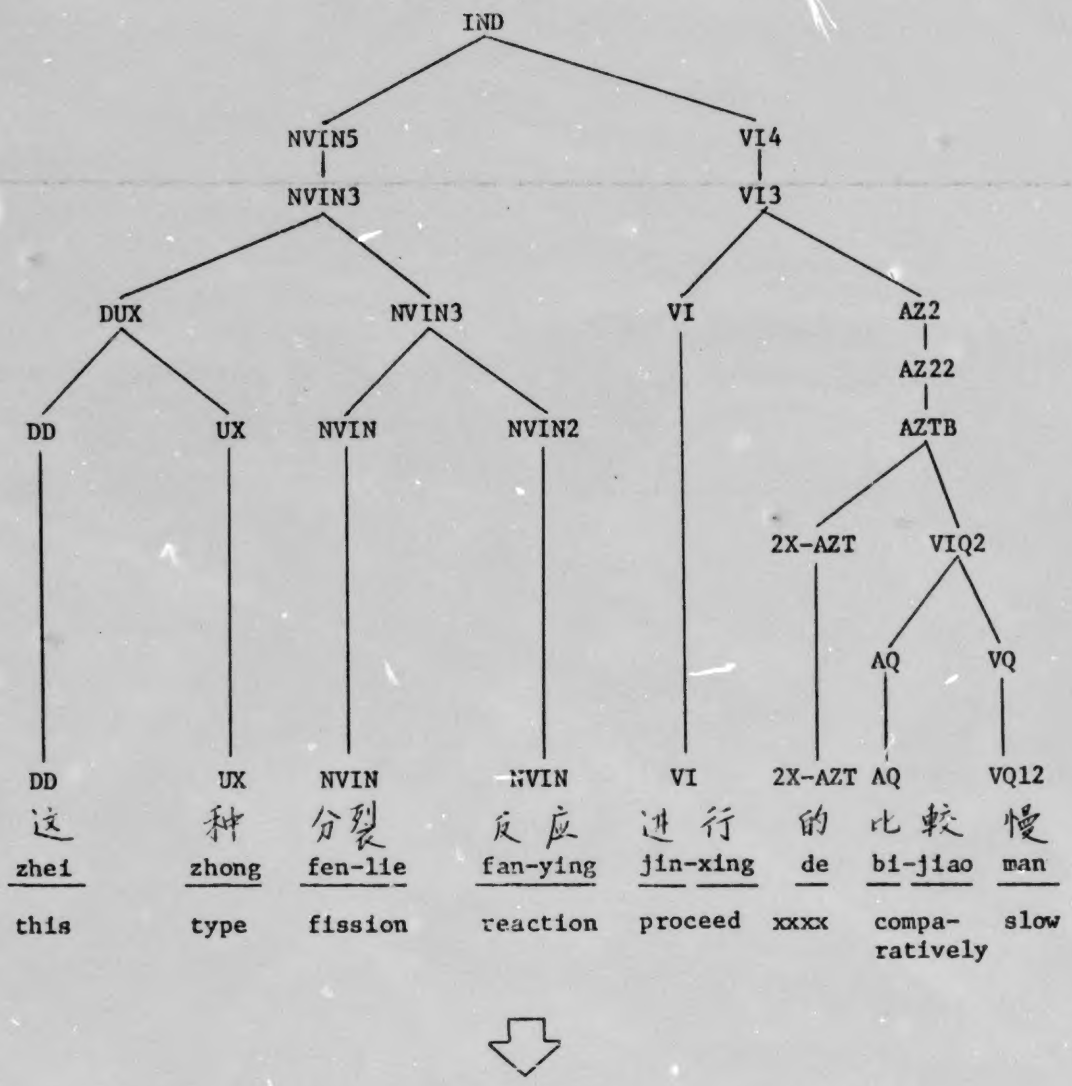

IND
NB5
NB3
NB3*R
NDEO*R
NB2
NN5
NN
NC
DE2
DE2
NB
VIQ*1
VIQ2
VIQ13*R
AQ2
WQ
NDE7*R
NN5
NN
NC
DE7
DE7
VIQ12*R
VQ12*2
VQ12
MUY
MMULT
MM5
MNC10
MTIM
MTIM
鐳 lei radium
的 de of
原子量 yuan-zi-liang atomic+weight
比 bi than
鈉 na sodium
的 de that+of
大 da large
十 shi ten
倍 bei times

'atomic weight of radium (be) ten times large(r) than that of sodium'

Interlingual operations:

(a) NDEO*R permutes NN5 'radium' and DE2 'of'.
(b) NB3*R permutes NDEO*R 'of radium' and NB2 'atomic+weight'.
(c) NDE7*R permutes NN5 'sodium' and DE7 'that+of'.
(d) VIQ12*R permutes VQ12*2 'large' and MUY 'ten times'.
(e) VIQ13*R permutes AQ2 'than that of sodium' and VIQ12*R '(be) ten times large(r)'.

Figure 14:


IND
NVIN5
VI4
NVIN3
VI3
DUX
NVIN3
VI
AZ2
AZ22
AZTB
DD
UX
NVIN
NVIN2
2X-AZT
VIQ2
AQ
VQ
DD
UX
NVIN
NVIN
VI
2X-AZT
AQ
VQ12
这 种 分裂 反应 进行 的 比較 慢
zhei
zhong
fen-lie
fan-ying
jin-xing
de
bi-jiao
man
this
type
fission
reaction
proceed
xxxx
compa-ratively
slow

'This type fission reaction proceed comparatively slow'

Interlingual operations: none

(B) Locational nouns and verbs

Lexical Types:

在 0961 *zai*	VG11	be+at/be+in	
	VG12	be	(Locative verbs)
	VG13	at/in	
	VG14	xxxx	
外 面 1120 7240 *wai-mian*	NL11	outside	
	NL12	outside+of	(Locative nouns)
	NL13	the+outsides	
	NL14	the+outsides+of	

A locative phrase in Chinese consists of at least two of the elements locative verb (VG), locative noun (NL), and noun other than NL (N), in the following structural combinations: (a) VG N, (b) VG N NL, (c) N NL, and (d) VG NL, where *de*/*zhi* can optionally occur between N and NL. Structures (a), (b), and (d) can function as predicates and adverbials, (c) as a noun phrase, and all four structures as noun modifiers. The co-occurrence of N and NL is dependent on the semantic characteristics of N: where N itself has certain locative qualities (e.g. names of buildings or other very large structures, names of cities, countries, and other geographic features, etc.), and NL may be optional or even disallowed if its meaning can be predicted from the combination VG + N. In all other cases, NL is obligatory.

Where a locative phrase functions as a predicate, VG is coded VG12 'be' if an NL is present, or VG11 'be at/in/on' if NL is absent. Where a locative phrase functions as adverb or noun modifier, VG is rendered as VG14 'xxxx' or VG13 'at/in/on', again depending on the presence or absence of NL. In adverbial locative phrases, NL is glossed as a preposition (NL12) where N is present or as a locative adverb (NL11) where N is absent. Where a locative phrase acts as a a nominal, NL is rendered as a nominalized preposition (NL13), or nominalized preposition + 'of' when accompanied by N.

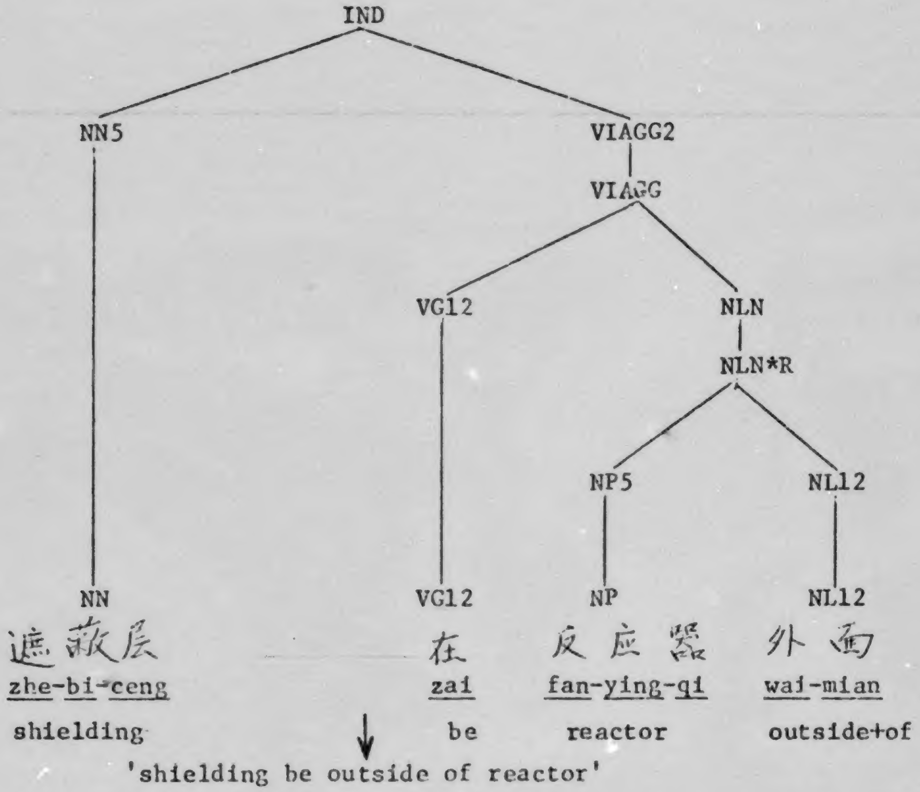
Wherever N and NL co-occur, they are permuted interlingually via the node NLN*R. Any *de/zhi* intervening between N and NL is coded DE4 so that no permutation of N and *de/zhi* is necessary. All locative phrases as adverbials are eventually labelled NLN if they occur pre-verbally, AGG22 if they occur post-verbally. Since English requires that all locative adverb phrases dominated by a predicate *follow* the verb, all sequences consisting of NLN + higher verb-phrase (e.g. VI3) are permuted interlingually via a *R node (e.g. VI4*R). AGG22, however,is not permuted since it is already in a suitable position following the main verb.

The top node for all locative phrases functioning as predicates is VIAGG22. When locative structural types (a), (b) and (d) are used as noun modifiers, it is as VIAGG22 rather than NLN, VIAGG22 taking DE3/DE5/JY3/JY5 and undergoing the same interlingual permutations as any other embedded verb phrase. On the other hand, locative structural type (c) does embed as NLN, inasmuch as this is the only structural analysis possible; here, DE4 is specified and the resultant node NDAGG is permuted with the head noun.

Index of figures

Figure 15-16: NL12/NLN /VG12
17: NL11/VIAGG2/VG12
18: NL12/VIAGG2/VG12
19: NL12/NDAGG/NL12
20: NL12/AGG15/VG14
21: NL13/NLNN /VG14
22: NL14/NLNN /NL13
23: NL12/AGG15/NGG /VG14

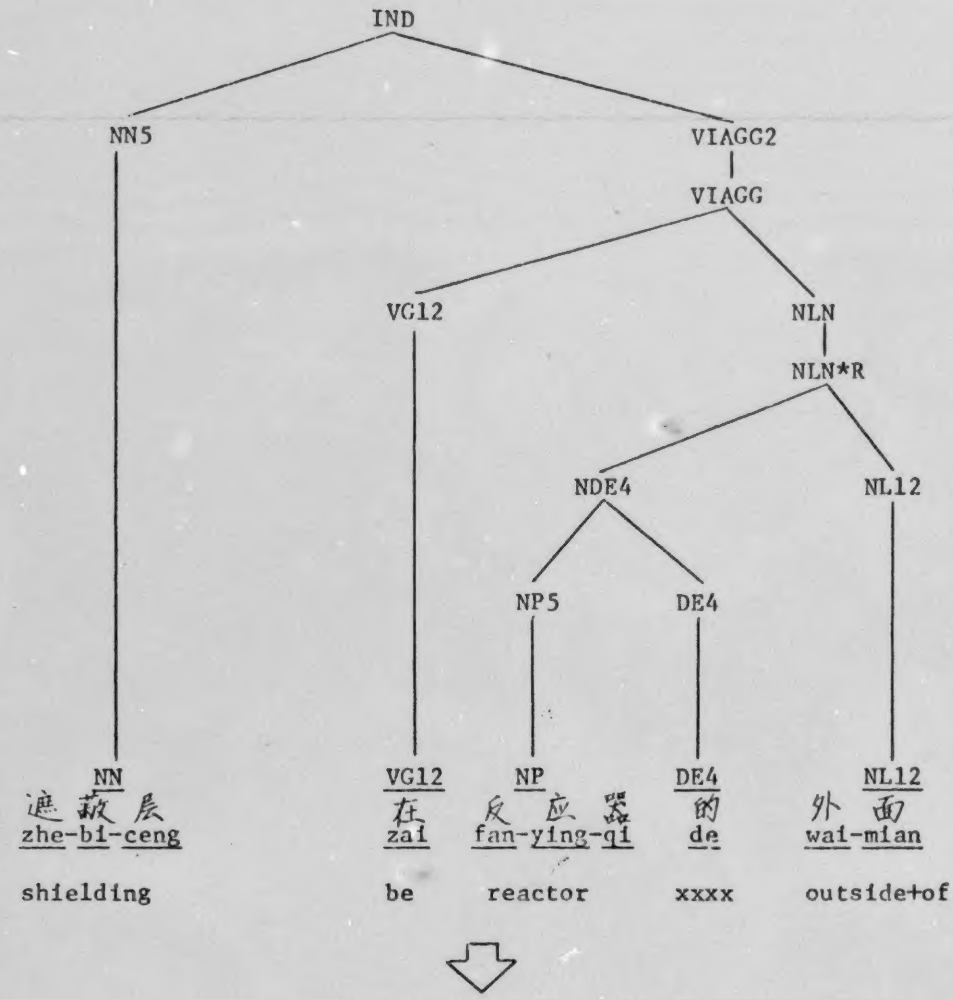
Figure 15:


IND
NN5
VIAGG2
VIAGG
VG12
NLN
NLN*R
NP5
NL12
NN
VG12
NP
NL12
遮蔽层
在
反应器
外面
zhe-bi-ceng
zai
fan-ying-qi
wai-mian
shielding
be
reactor
outside+of
'shielding be outside of reactor'

Interlingual operations:

NLN*R permutes NP5 'reactor' and NL12 'outside of'.

Figure 16:

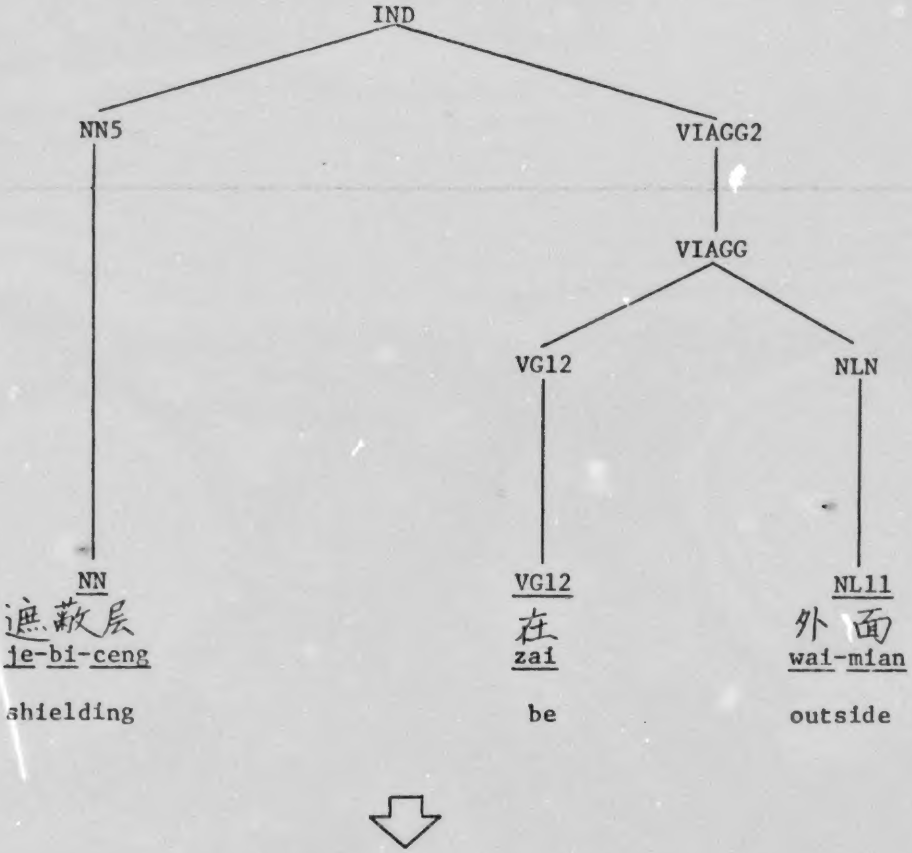

IND
NN5
VIAGG2
VIAGG
VG12
NLN
NLN*R
NDE4
NL12
NP5
DE4
NN
VG12
NP
DE4
NL12
遮蔽层
zhe-bi-ceng
在
zai
反应器
fan-ying-qi
的
de
外面
wai-mian
shielding
be
reactor
xxxx
outside+of

'shielding be outside of reactor'

Interlingual operations:

NLN*R permutes NDE4 'reactor' with NL12 'outside of'.

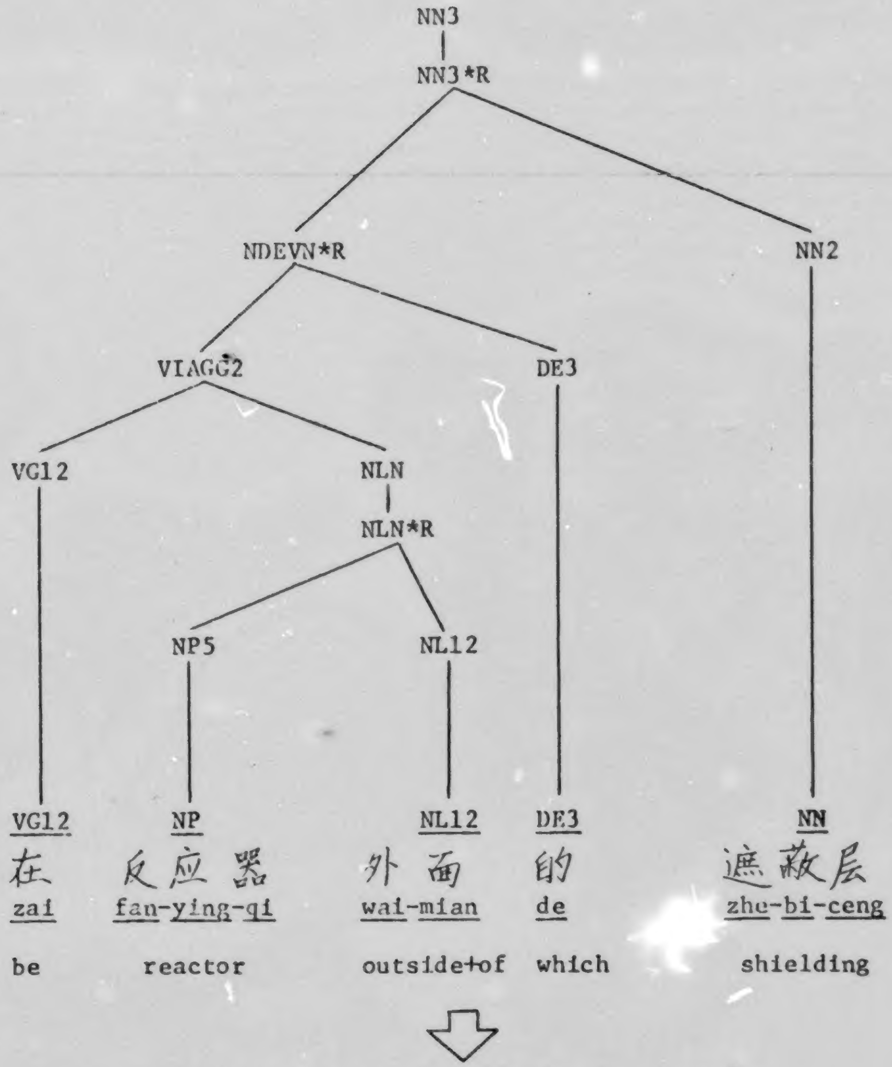
Figure 17:


IND
NN5
VIAGG2
VIAGG
VG12
NLN
NN
遮蔽层
je-bi-ceng
shielding
VG12
在
zai
be
NL11
外面
wai-mian
outside

'shielding be outside'

Interlingual operations: none

Figure 18:

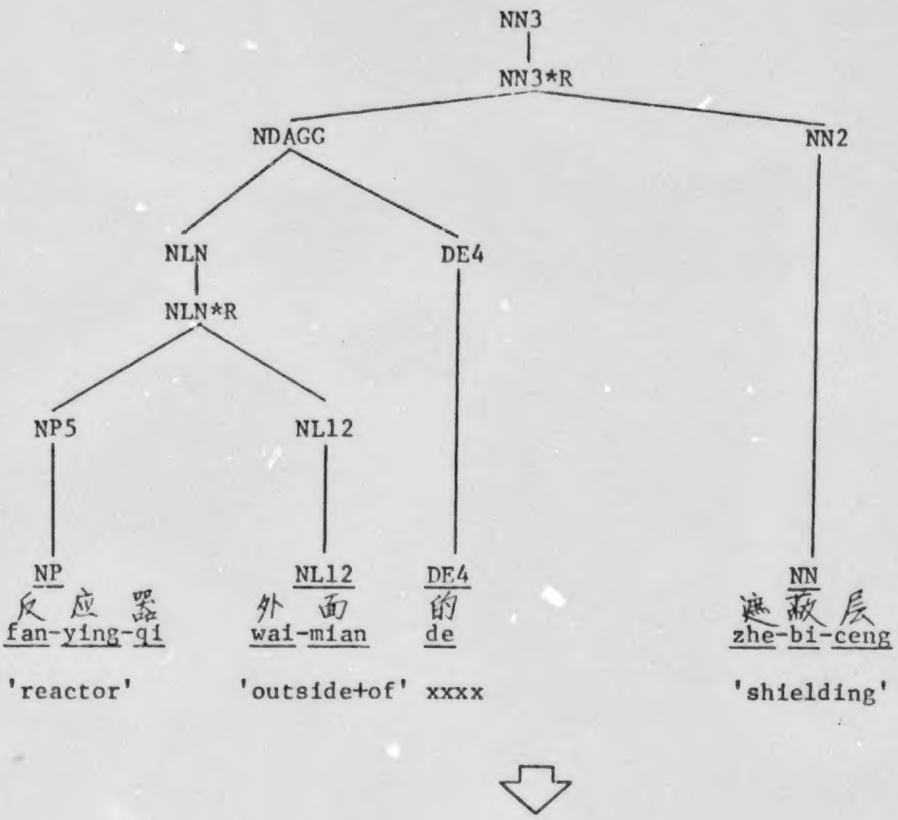

NN3
NN3*R
NDEVN*R
NN2
VIAGG2
DE3
VG12
NLN
NLN*R
NP5
NL12
VG12
NP
NL12
DE3
NN
在
反应器
外面
的
遮蔽层
zai
fan-ying-qi
wai-mian
de
zhe-bi-ceng
be
reactor
outside+of
which
shielding

'shielding which be outside of reactor'

Interlingual operations:

(a) NLN*R as in Figure 15.
(b) NDEVN*R permutes VIAGG2 'be outside of reactor' with DE3 'which'
(c) NN2*R permutes NDEVN*R 'which be outside of reactor' and NN2 'shielding'.

Figure 19:


NN3
NN3*R
NDAGG
NN2
NLN
DE4
NLN*R
NP5
NL12
NP
NL12
DE4
NN
反应器
fan-ying-qi
外面
wai-mian
的
de
遮蔽层
zhe-bi-ceng
'reactor'
'outside+of'
xxxx
'shielding'

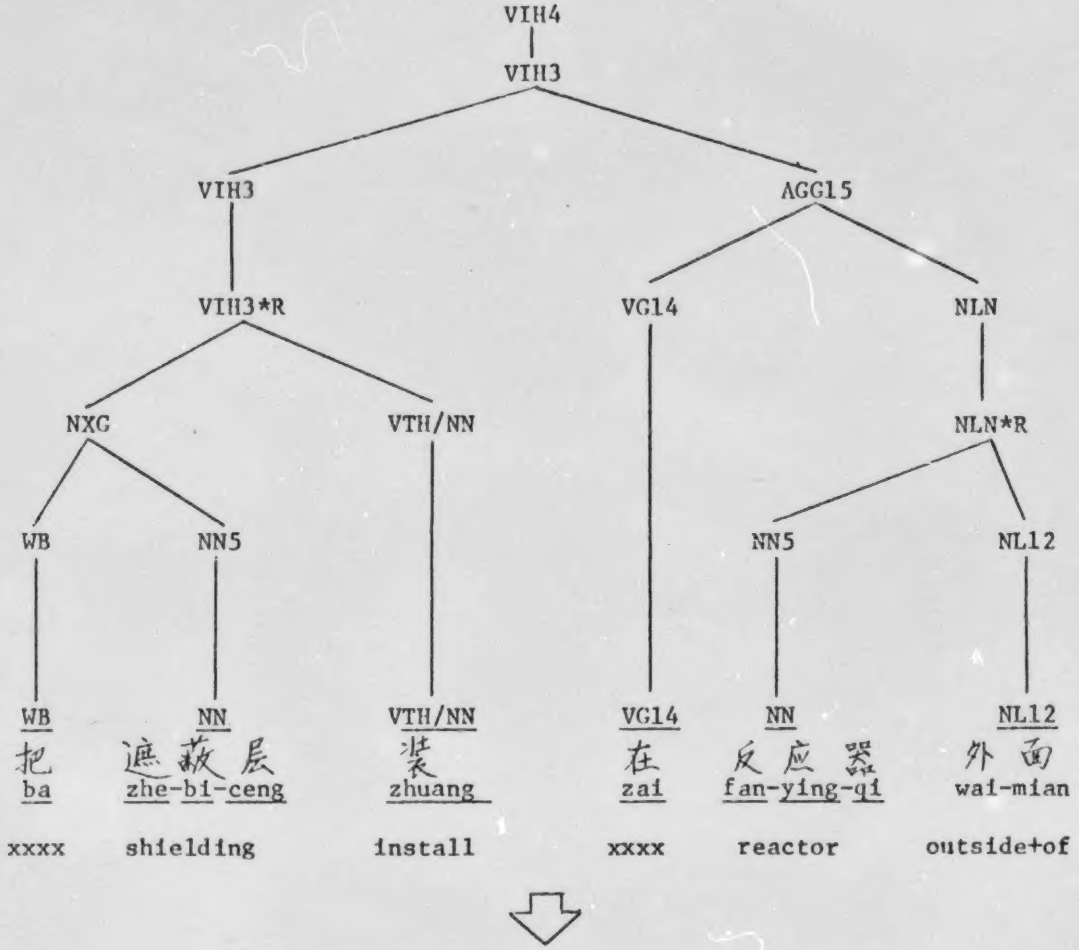
'shielding outside of reactor'

Interlingual operations:

(a) NLN*R as in Figure 15.

(b) NN3*R permutes NDAGG 'outside of reactor' and NN2 'shielding'.

Figure 20:


VIH4
VIH3
VIH3
AGG15
VIH3*R
VG14
NLN
NXG
VTH/NN
NLN*R
WB
NN5
NN5
NL12
WB
NN
VTH/NN
VG14
NN
NL12
把
遮蔽层
装
在
反应器
外面
ba
zhe-bi-ceng
zhuang
zai
fan-ying-qi
wai-mian
xxxx
shielding
install
xxxx
reactor
outside+of

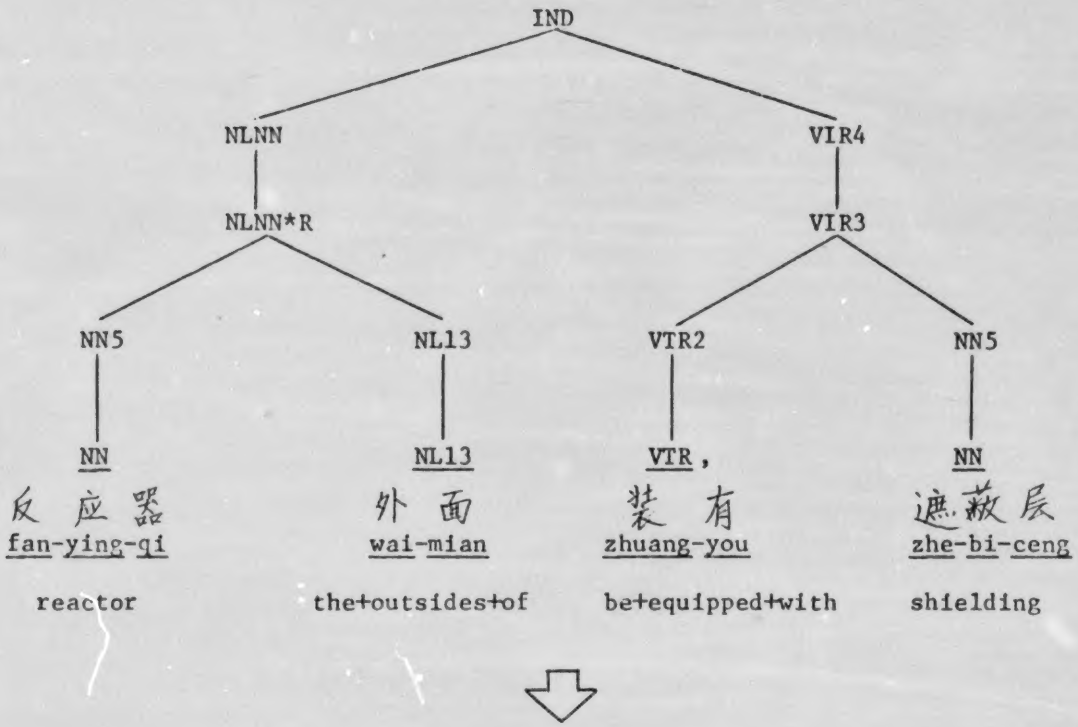
'install shielding outside of reactor'

Interlingual operations:

(a) NLN*R as in Figure 15.

(b) VIH3*R permutes NXG 'shielding' and VTH/NN 'install'.

Figure 21:

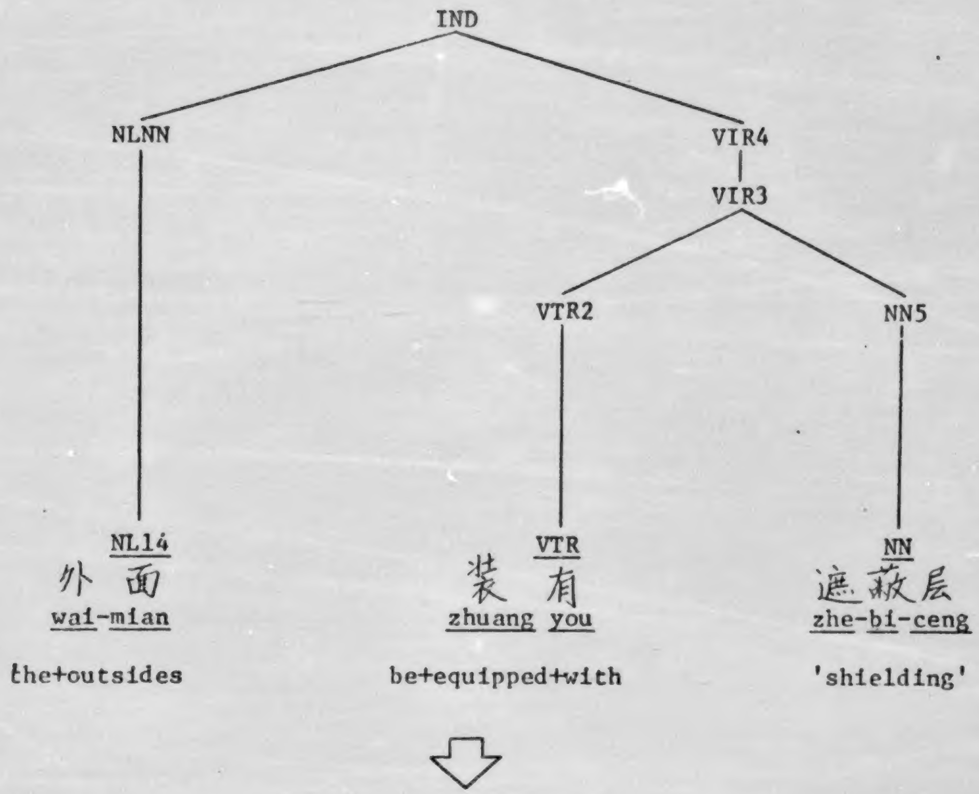

IND
NLNN
VIR4
NLNN*R
VIR3
NN5
NL13
VTR2
NN5
NN
NL13
VTR,
NN
反 应 器
fan-ying-qi
reactor
外 面
wai-mian
the+outsides+of
装 有
zhuang-you
be+equipped+with
遮蔽层
zhe-bi-ceng
shielding

'the outsides of reactor be equipped with shielding'

Interlingual operations:

NLNN*R permutes NN5 'reactor' and NL13 'the outsides of'.

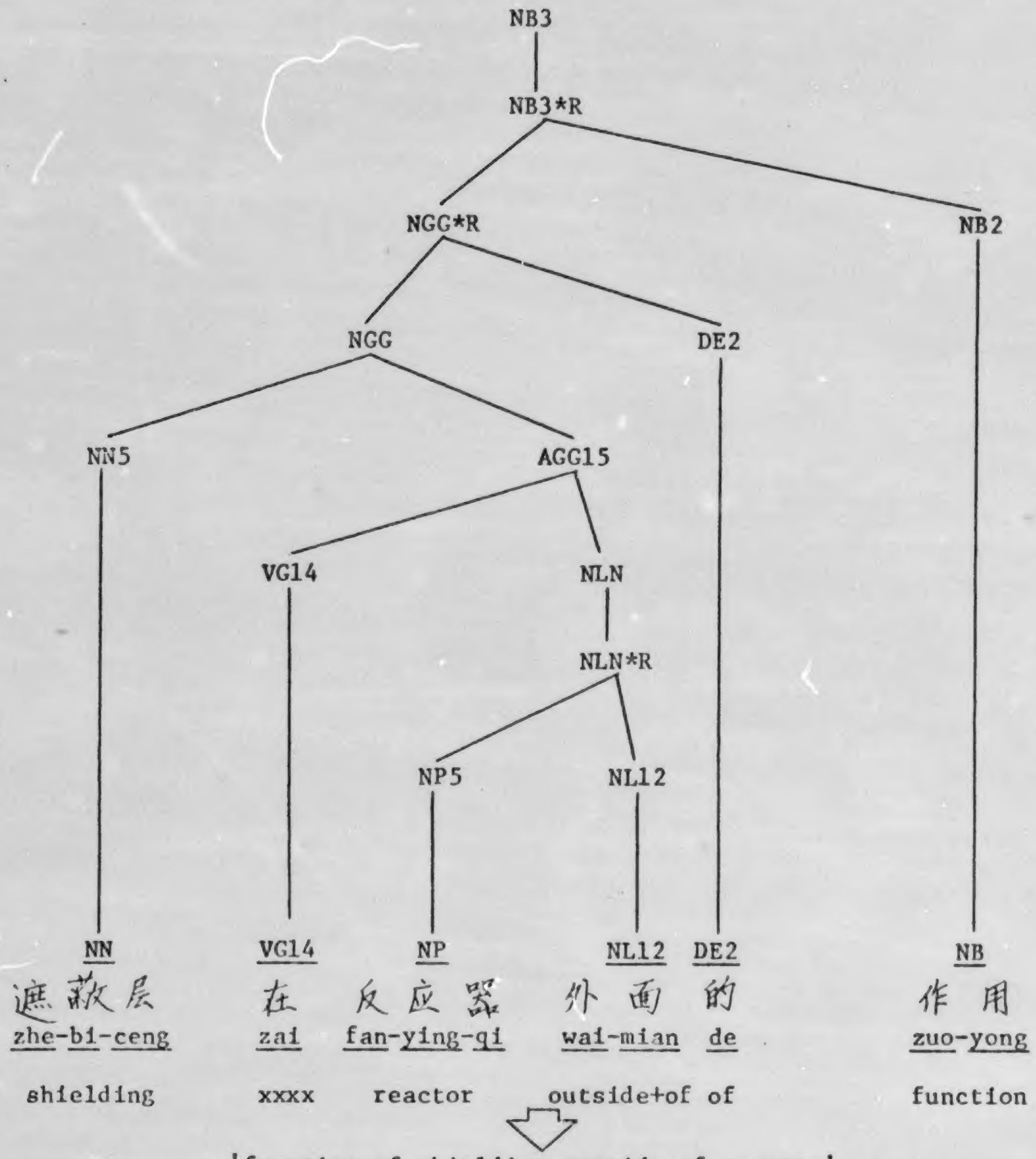
Figure 22:


IND
NLNN
VIR4
VIR3
VTR2
NN5
NL14
外 面
wai-mian
the+outsides
VTR
装 有
zhuang you
be+equipped+with
NN
遮蔽层
zhe-bi-ceng
'shielding'

'the outsides be equipped with shielding'

Interlingual operations: none.

Example 23:


NB3
NB3*R
NGG*R
NB2
NGG
DE2
NN5
AGG15
VG14
NLN
NLN*R
NP5
NL12
NN
VG14
NP
NL12
DE2
NB
遮蔽层
在
反应器
外面
的
作用
zhe-bi-ceng
zai
fan-ying-qi
wai-mian
de
zuo-yong
shielding
xxxx
reactor
outside+of
of
function

'function of shielding outside of reactor'

Interlingual operations: (a) NLN*R as in Figure 15.
(b) NGG*R permutes NGG 'shielding outside of reactor' and DE2 'of'
(c) NB3*R permutes NGG*R 'of shielding outside of reactor' and NB2 'function'.

(C) *Directional verbs and complements* (cf. Chao 1968:458-67, Fan 1963).

Lexical types:

来	0171	VD11	come
	lai	VD12	xxxx
去	0637	VD11	go
	qu	VD12	xxxx
上	0006	VD13	rise/ascend
	shang	VD14	up
下	0007	VD13	fall/descend
	xia	VD14	down
出	0427	VD13	exit/go+out
	chu	VD14	out
进	6651	VD13	enter
	jin	VD14	in
过	6665	VD13	pass
	guo	VD14	over
回	0932	VD13	return
	hui	VD14	back

The above lexical items, which we have termed "directionals", can oocur as main verbs of predicates, or as complements to other verbs. They are divided into two groups: (a) VD11-12, indicating motion towards (来 *lai*) or away from (去 *qu*) the speaker, and (b) VD13-14, verbs indicating more specific directional components, e.g. motion up, down, in, out, etc. When *lai* and *qu* occur alone as main verbs, they are glossed VD11 'come' and 'go', respectively. Group (b) items can also participate in directional-verb predicates, where they are coded VD13 and receive appropriate gloss equivalents as English verbs. Such items cannot occur in isolation, but must be complemented either by post-posed VD12 lai or qu, or else by nouns which are intrinsically locative, e.g. 北京 bei-jing 'Peking' or 美国 mei-guo 'United States'.

Group (a) items, (b) items, or (b) + (a) items in combination may be post-posed as directional complements to other verbs of motion. As complements are coded VD14 and receive appropriate glosses as English post-verbal adverbs. Note however, that lai and qu as complements are always given zero glosses (VD12). This is because in English such notions as "motion toward the speaker" and "motion away from the speaker" in directional complements are not expressed by any one or two particular lexical items which could be equivalenced to lai or qu; but are normally gauged from the context surrounding the complement instead. Inasmuch as this context is to a great extent redundantly supplied by the Chinese text as it is translated, we are justified in assigning complemental lai and qu null glosses to begin with rather than resorting to any more elaborate treatment.

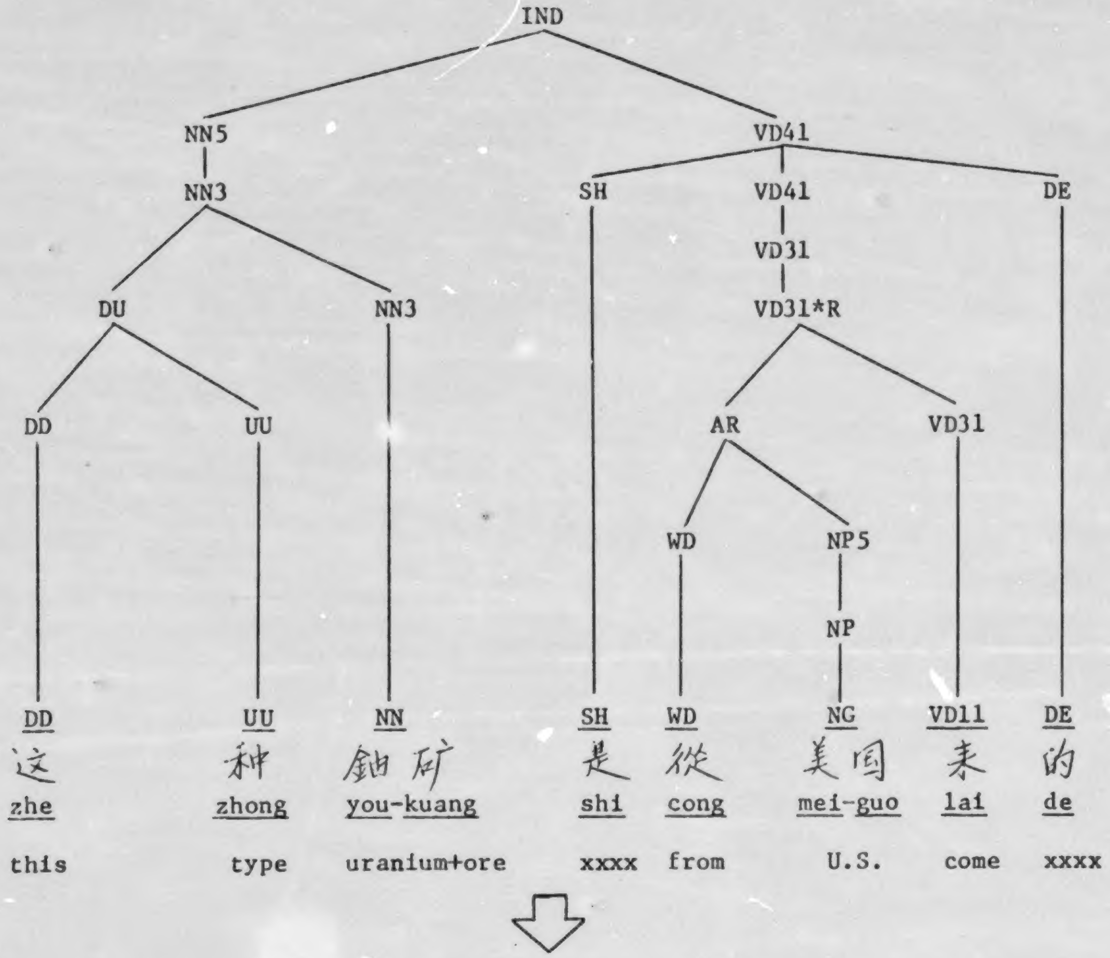
In the following figures, predicates containing directionals as main verbs have VD41 as their top node, while directional ocomplements are assigned the top node AZDR.

Index to Figures:

Figure 24: VD11/VD41
25: VD13/VD11/VD41
26: VD12/VD14/AZDR

Note that in Figure 26, AZDR < VD12+VD14 may optionally be replaced by an AZDR consisting of either VD12 or VD14 alone, with accompanying change in meaning.

Figure 24:

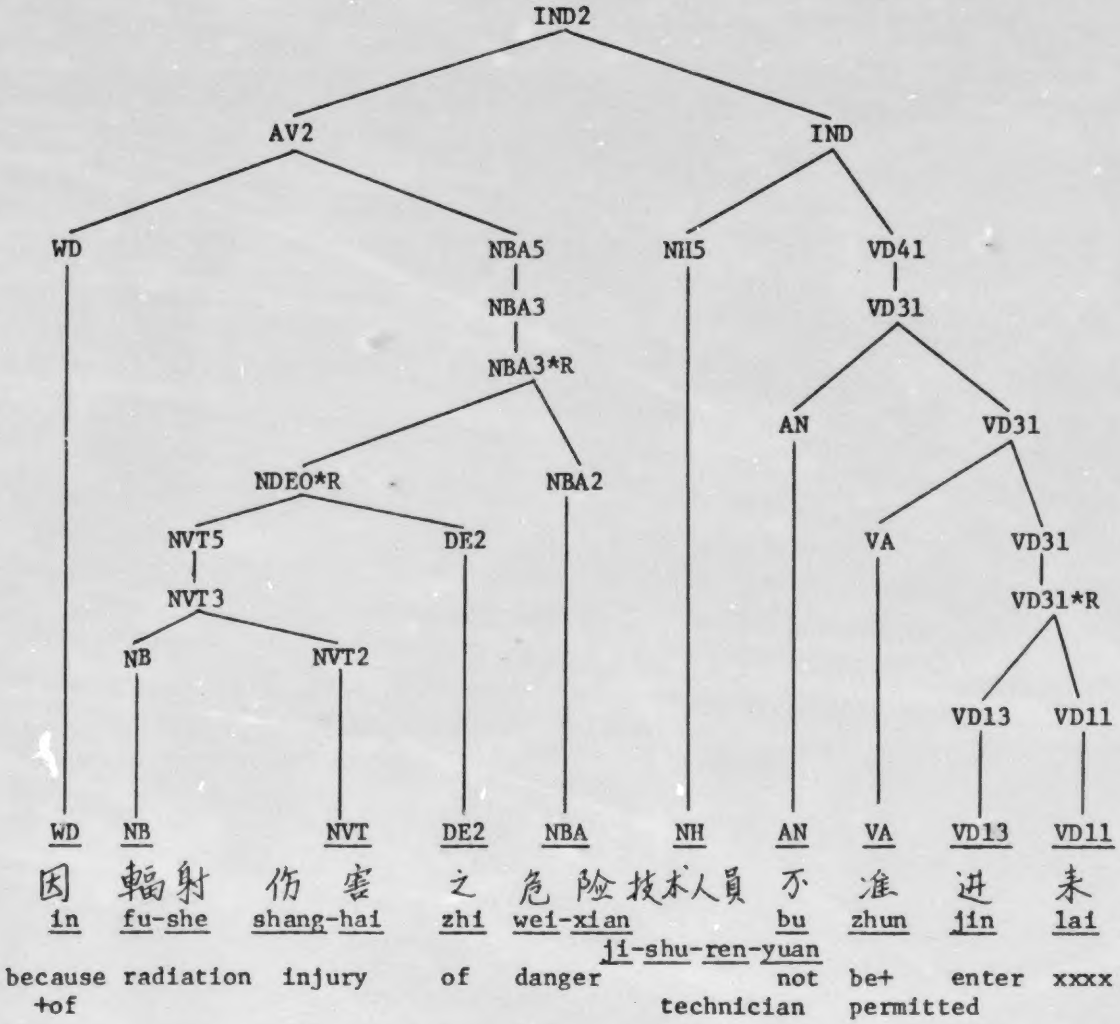

IND
NN5
VD41
NN3
SH
VD41
DE
VD31
DU
NN3
VD31*R
DD
UU
AR
VD31
WD
NP5
NP
DD
UU
NN
SH
WD
NG
VD11
DE
这
种
鈾矿
是
從
美国
来
的
zhe
zhong
you-kuang
shi
cong
mei-guo
lai
de
this
type
uranium+ore
xxxx
from
U.S.
come
xxxx

'this type uranium ore come from U.S.'

Interlingual operations:

VD31*R permutes AR 'from U.S.' and VD31 come'.

Figure 25:

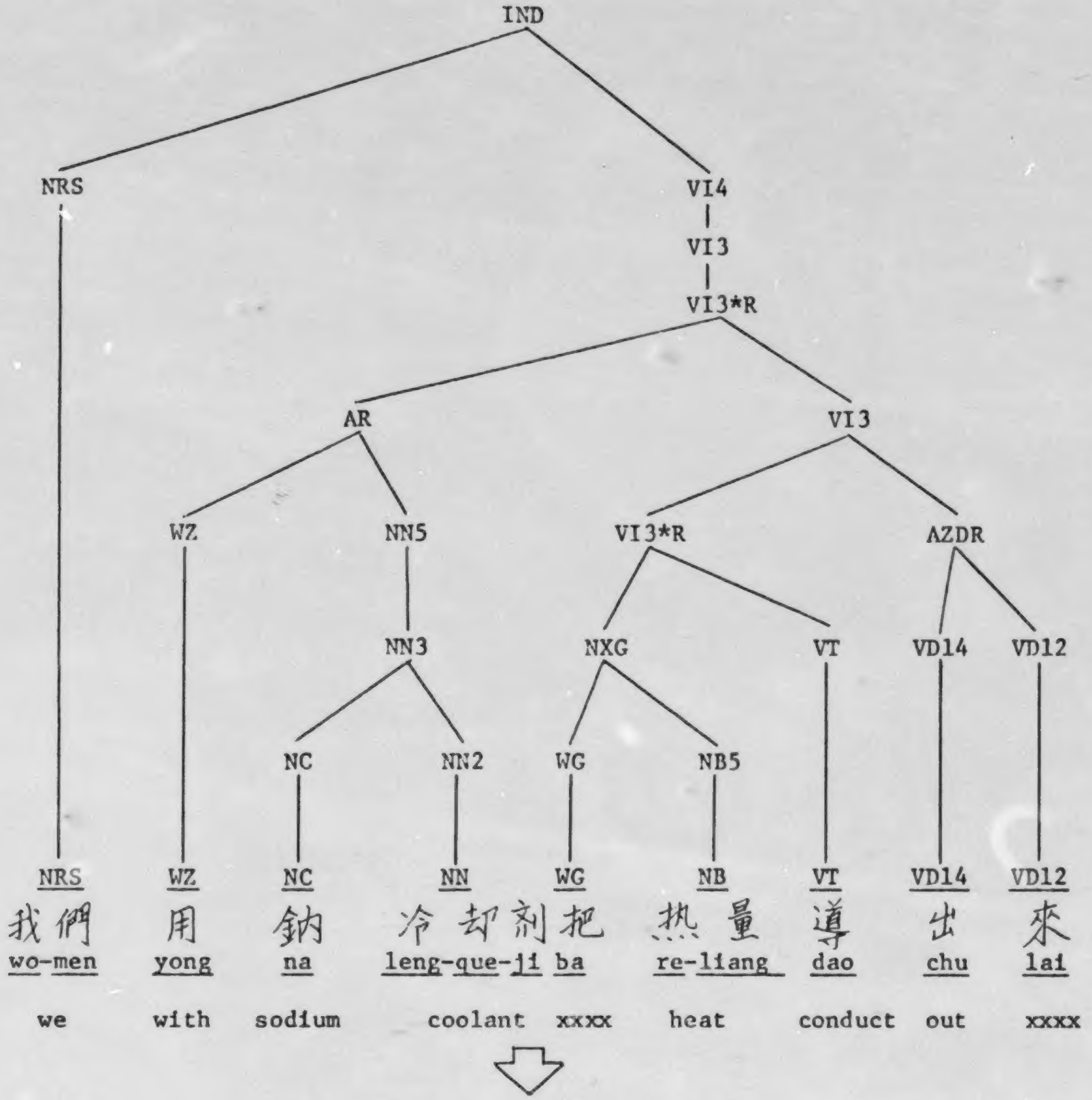

IND2
AV2
IND
WD
NBA5
NH5
VD41
NBA3
VD31
NBA3*R
AN
VD31
NDEO*R
NBA2
NVT5
DE2
VA
VD31
NVT3
VD31*R
NB
NVT2
VD13
VD11
WD
NB
NVT
DE2
NBA
NH
AN
VA
VD13
VD11
因 輻射 伤害 之 危险 技术人員 不 准 进 来
in fu-she shang-hai zhi wei-xian bu zhun jin lai
ji-shu-ren-yuan
because +of
radiation
injury
of
danger
not
technician
be+ permitted
enter
xxxx

'because of danger of radiation injury, technician not be permitted enter'.

Interlingual

(a) NDEO*R permutes NVT5 'radiation injury' and DE2 'of'.
(b) NBA3*R permutes NDEO*R 'of radiation injury' and NBA2 'danger'.

Figure 26:


IND
NRS
VI4
VI3
VI3*R
AR
VI3
WZ
NN5
VI3*R
AZDR
NN3
NXG
VT
VD14
VD12
NC
NN2
WG
NB5
NRS
WZ
NC
NN
WG
NB
VT
VD14
VD12
我們
用
鈉
冷却剂
把
热量
導
出
來
wo-men
yong
na
leng-que-ji
ba
re-liang
dao
chu
lai
we
with
sodium
coolant
xxxx
heat
conduct
out
xxxx

'we conduct heat out with sodium coolant'

Interlingual operations:
- (a) VI3*R (lower) permutes NXG 'heat' and VT'conduct'.
- (b) VI3*R (upper) permutes AR 'with sodium coolant' and VI3 'conduct heat out'.

(D) Function words 以 yi, 為 wei, and 作 zuo

Lexical types:

以
0110
yi
- CV in+order+to
- VTWE13 consider
- VTWE14 use
- WD with
- 1VAK-VA xxxx

為
3634
wei
- CV in+order+to
- VC be
- WD for/for+the+purpose+of
- WE11 be
- WE12 into
- WE13 as
- WP12 xxxx

作
0155
zuo
- VTH/NB do/perform
- VTH/NH be/act+as/serve+as
- VTH/NN make
- WE11 be
- WE13 as

In classical Chinese, the lexical items yi/wei/zuo were once verbs: yi meaning 'to take', wei 'to be, act as, make' and zuo 'to act as, make'. The situation in Modern Chinese is far more complex, as can easily be gauged from the lexical listings above. Not only are the original meanings of yi, wei, and zuo retained, but new meanings and grammatical functions have evolved. In particular, WD yi plus noun can now act as a pre-verbal instrumental adverb phrase, and both CV yi and wei can function as subordinating conjunctions introducing purpose clauses. WD wei also indicates purpose in prepositional phrases, and can furthermore act as a passive marker. Finally, WE11-13 wei and zuo for post-verbal complements fulfilling a variety of semantic roles. Not surprisingly, this diversification of functions can sometimes result in ambiguous interpretations, as shown by Figures 29 and 30, where *zuo* could be analyzed either as VTH/NN 'make' or WE13 'as', the two possible structures being quite different in meaning. Another interesting case is the string 可以 *ke-yi*, which depending on the context can be interpreted either as the lexical item *ke-yi* VA 'be able' or as the sequence *ke* VAK (also meaning 'be able') plus *yi* in any one of the functions listed above except CV.

In the following structural diagrams, note that CV and WD *wei* can be replaced by the homophonous derived forms 為了 *wei-le* and 為着 *wei-zhe*; also, *wei* and *zuo* are in free variation as WE11 and WE13.

Index to figures:

Figure 27: CV
28: VTWE13/WE11
29: VTWE14/WE13
30: WD *yi*/VTH/NN
31: 1VAK-VA *yi*/WE13
32: VC *wei*
33: WD *wei*
34: VTWE13/WE11
35 -36: VTWE11/WE11
37- 38: WE12
39: VIWE23/WE13
40: VTH/NH *zuo*

Figure 27:

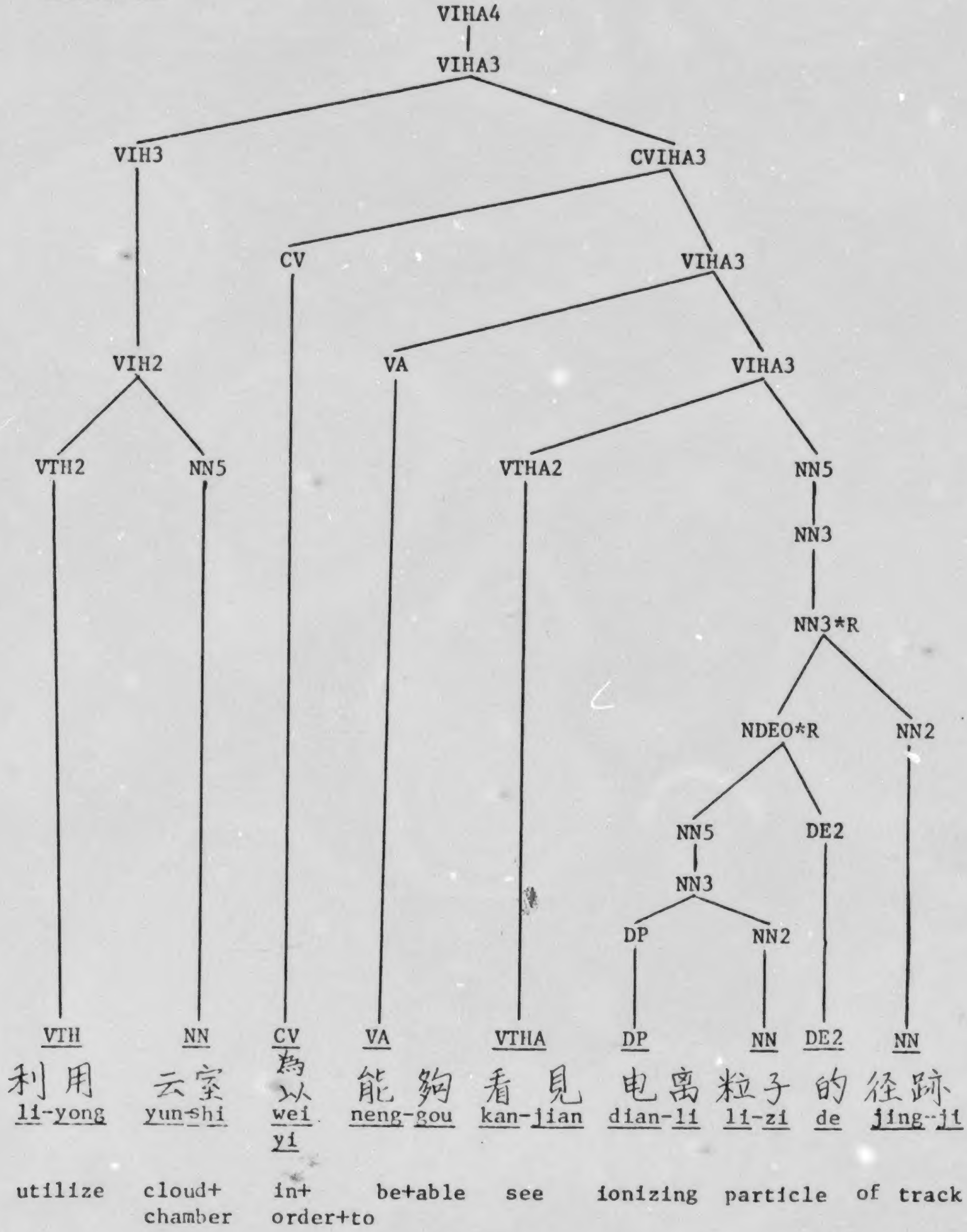

VIHA4
VIHA3
VIH3
CVIHA3
CV
VIHA3
VIH2
VA
VIHA3
VTH2
NN5
VTHA2
NN5
NN3
NN3*R
NDEO*R
NN2
NN5
DE2
NN3
DP
NN2
VTH
NN
CV
VA
VTHA
DP
NN
DE2
NN
利用
云室
為以
能夠
看見
电离
粒子
的
径跡
li-yong
yun-shi
wei yi
neng-gou
kan-jian
dian-li
li-zi
de
jing-ji
utilize
cloud+ chamber
in+ order+to
be+able
see
ionizing
particle
of
track

Figure 27: (continued) ⇩

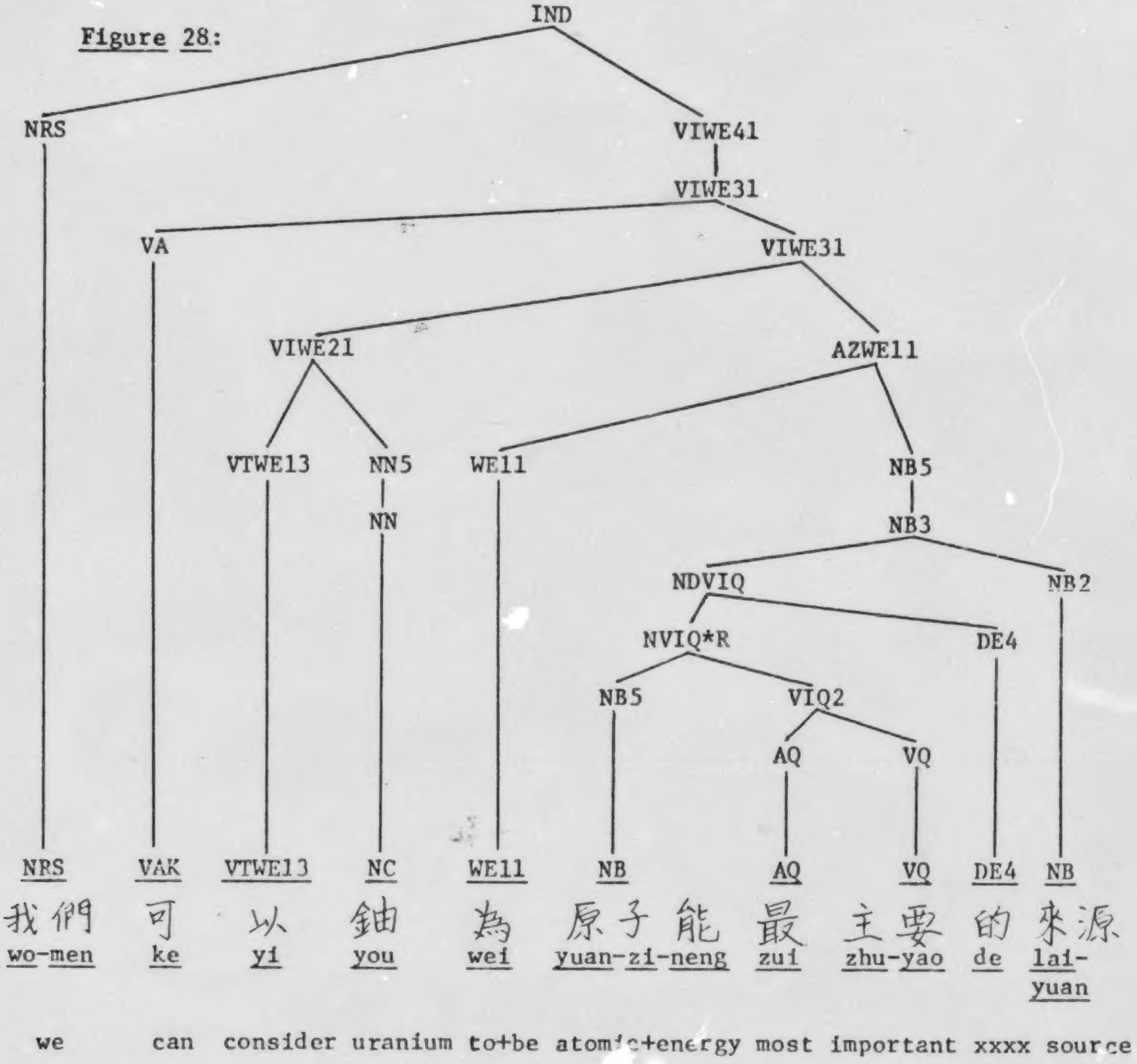
'utilize cloud chamber in order be able see **track** of ionizing particle'

Interlingual operations:

(a) NDEO*R permutes NN5 'ionizing particle' and DE2 'of'.

(b) NN3*R permutes NDEO*R 'of ionizing particle' and NN2 'track'.

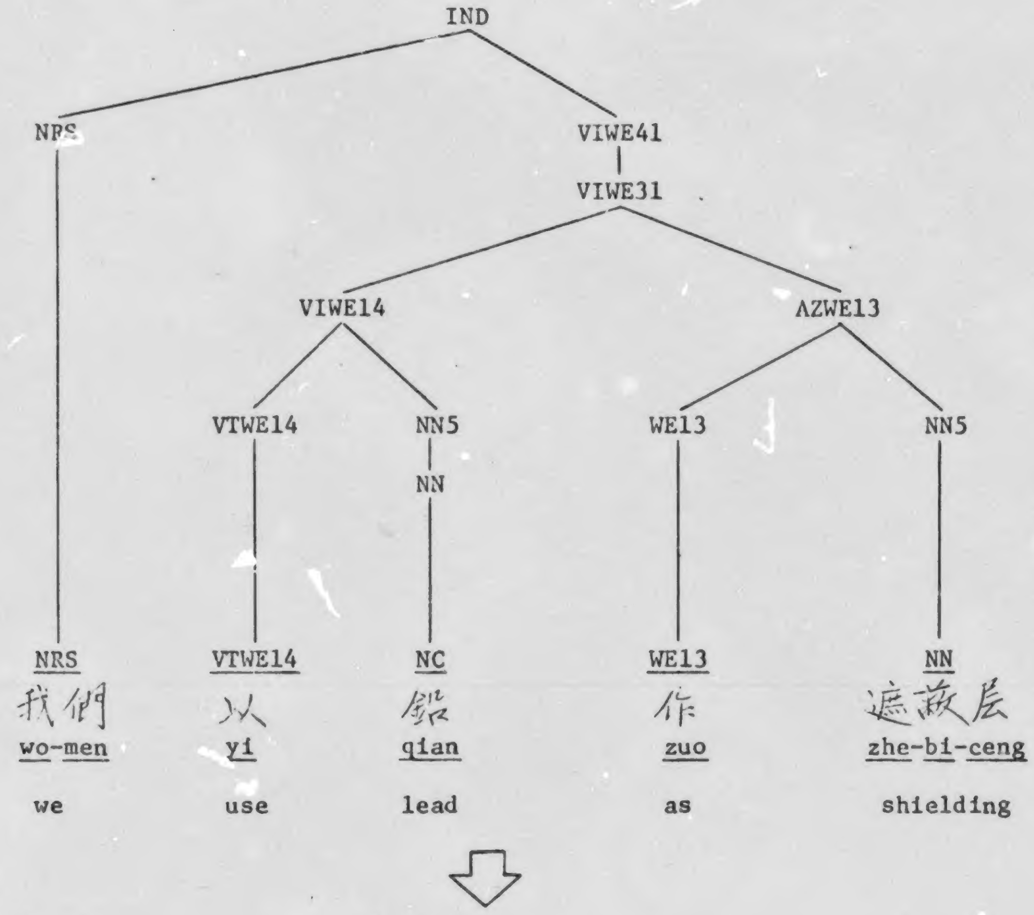
Figure 28:


IND
NRS
VIWE41
VIWE31
VA
VIWE31
VIWE21
AZWE11
VTWE13
NN5
NN
WE11
NB5
NB3
NDVIQ
NB2
NVIQ*R
DE4
NB5
VIQ2
AQ
VQ
NRS
VAK
VTWE13
NC
WE11
NB
AQ
VQ
DE4
NB
我們
可
以
鈾
為
原子能
最
主要
的
來源
wo-men
ke
yi
you
wei
yuan-zi-neng
zui
zhu-yao
de
lai-yuan
we
can
consider
uranium
to+be
atomic+energy
most
important
xxxx
source

'we can consider uranium to be most important atomic energy source'

Interlingual operations: NVIQ*R permutes NB5 'atomic energy' and VIQ2 'most important'.

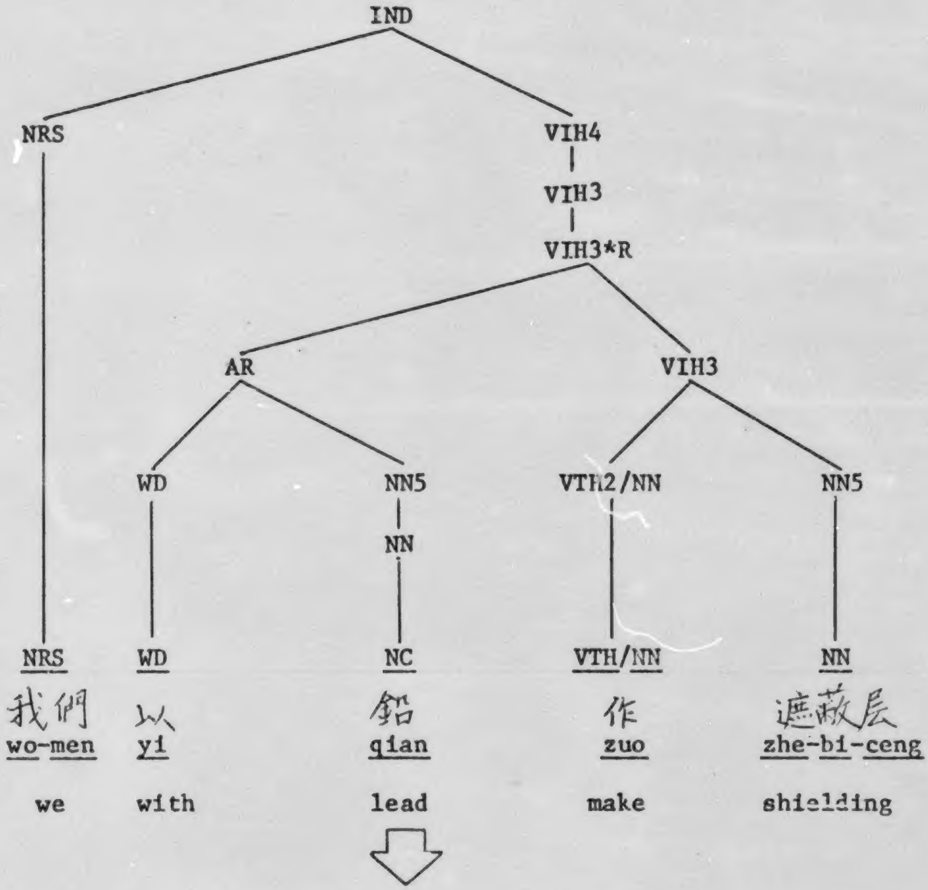
Figure 29:


IND
NRS
VIWE41
VIWE31
VIWE14
AZWE13
VTWE14
NN5
WE13
NN5
NN
NRS
VTWE14
NC
WE13
NN
我們
以
鉛
作
遮蔽层
wo-men
yi
qian
zuo
zhe-bi-ceng
we
use
lead
as
shielding

'we use lead as shielding'

Interlingual operations: none.

Figure 30:

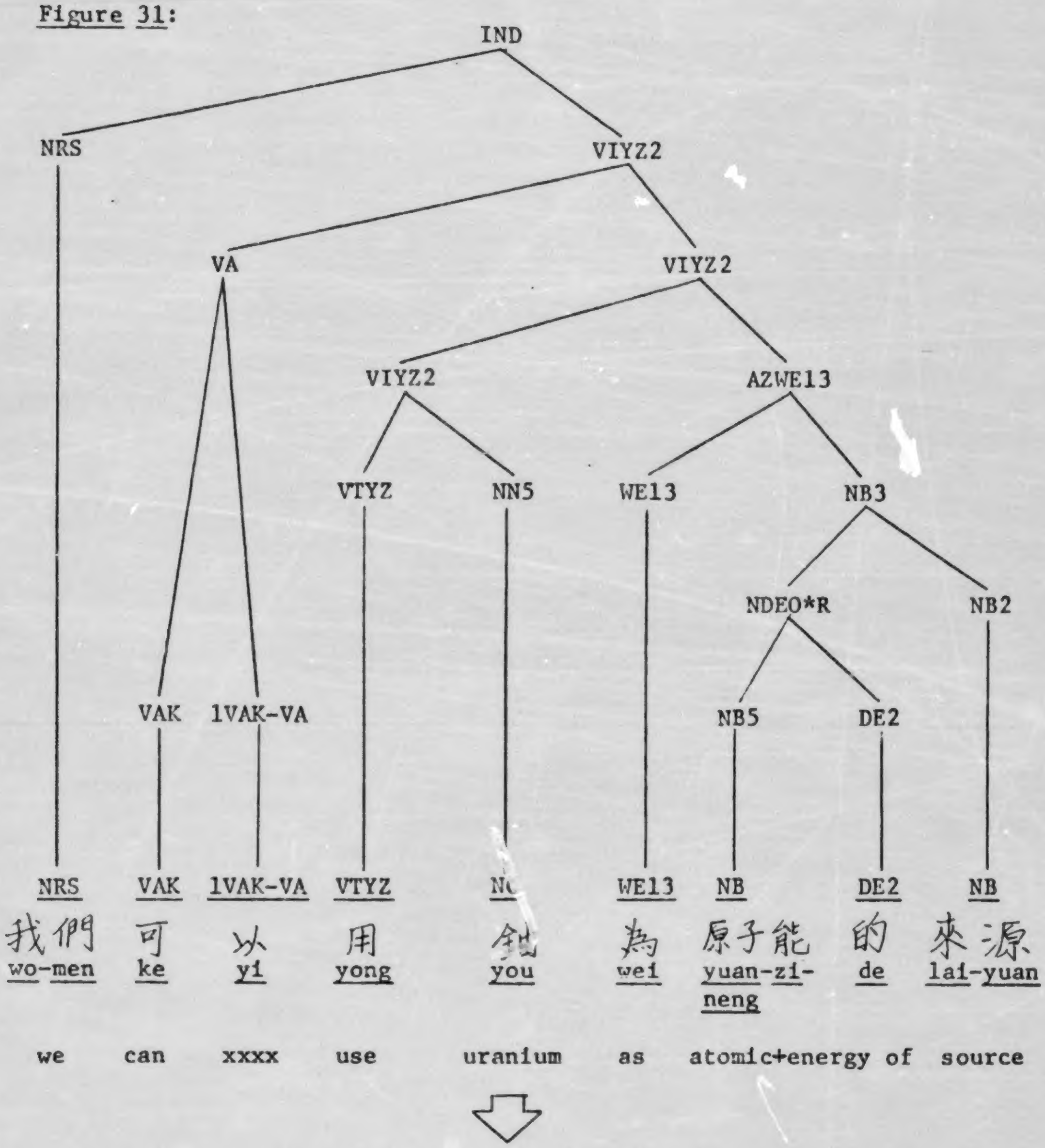

IND
NRS
VIH4
VIH3
VIH3*R
AR
VIH3
WD
NN5
VTH2/NN
NN5
NN
NRS
WD
NC
VTH/NN
NN
我們
wo-men
we
以
yi
with
鉛
qian
lead
作
zuo
make
遮蔽层
zhe-bi-ceng
shielding

'we make shielding with lead'

Interlingual

VIH3*R permutes AR 'with lead' and VIH3 'make shielding'.

Figure 31:

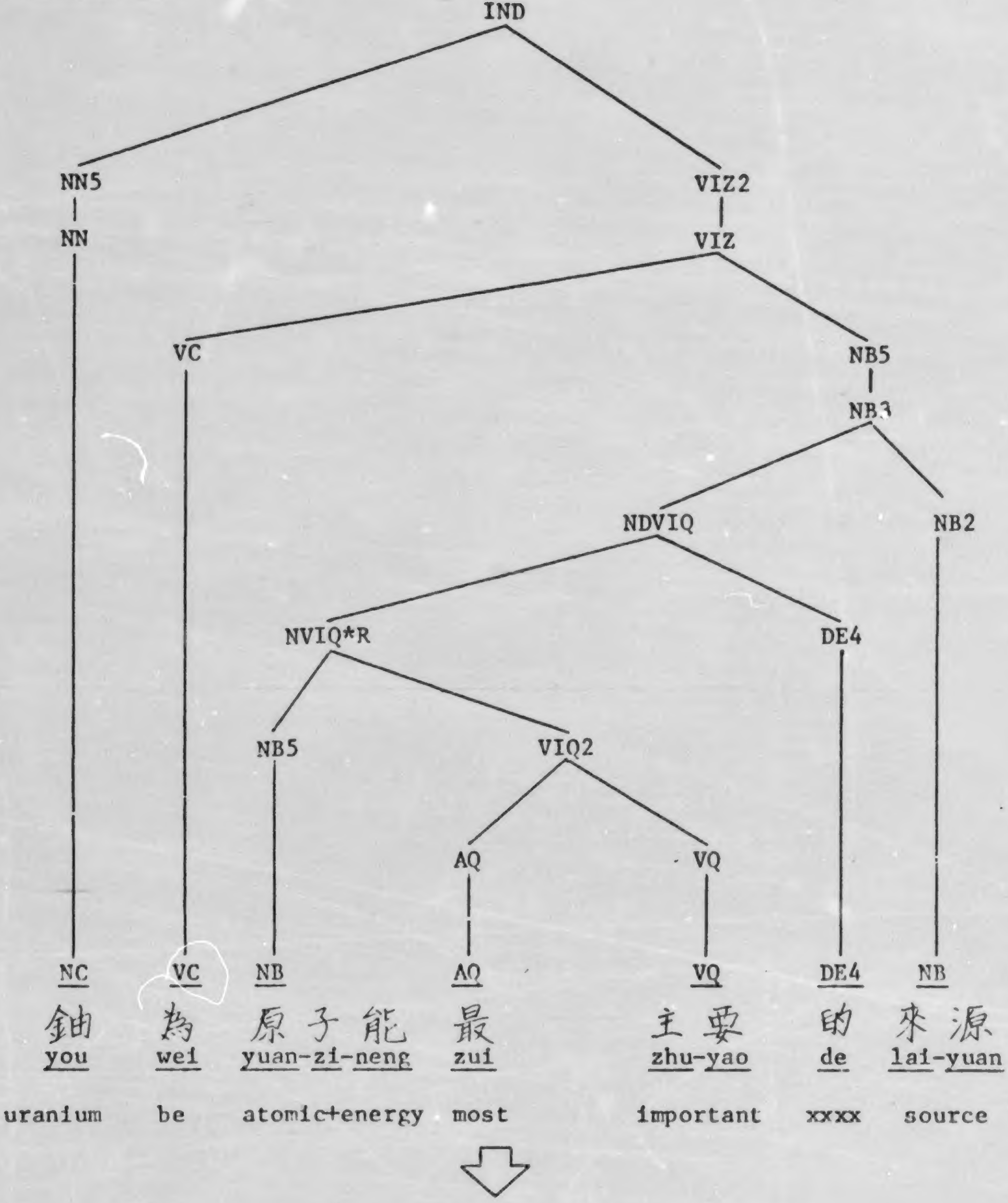

IND
NRS
VIYZ2
VA
VIYZ2
VIYZ2
AZWE13
VTYZ
NN5
WE13
NB3
NDEO*R
NB2
VAK
1VAK-VA
NB5
DE2
NRS
VAK
1VAK-VA
VTYZ
WE13
NB
DE2
NB
我們
可
以
用
鈾
為
原子能
的
來源
wo-men
ke
yi
yong
you
wei
yuan-zi-neng
de
lai-yuan
we
can
xxxx
use
uranium
as
atomic+energy
of
source

'we can use uranium as source of atomic energy'

Interlingual operations:

NDEO*R permutes NB5 'atomic energy' and DE2 'of'

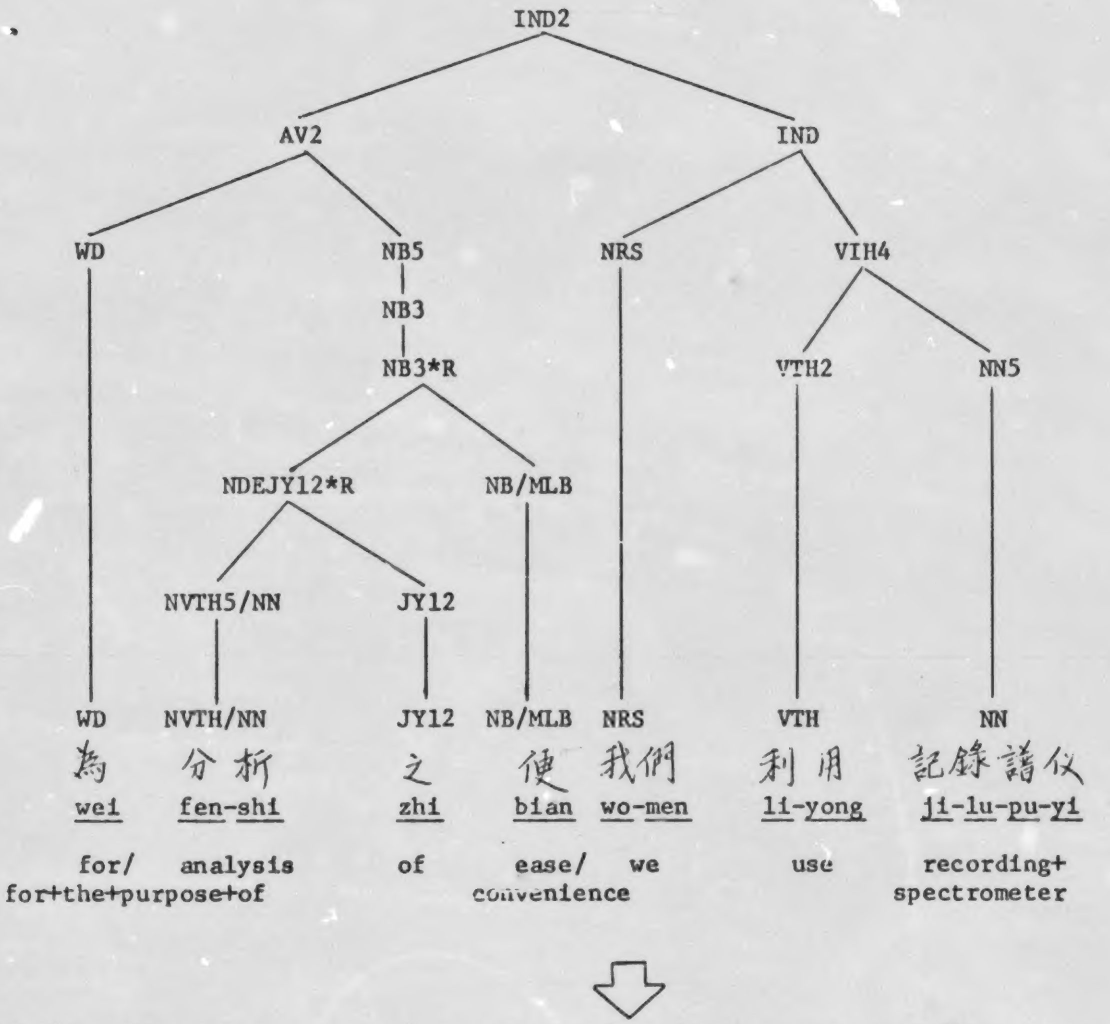
Figure 32:


IND
NN5
NN
VIZ2
VIZ
VC
NB5
NB3
NDVIQ
NB2
NVIQ*R
DE4
NB5
VIQ2
AQ
VQ
NC
VC
NB
AQ
VQ
DE4
NB
鈾
you
為
wei
原子能
yuan-zi-neng
最
zui
主要
zhu-yao
的
de
來源
lai-yuan
uranium
be
atomic+energy
most
important
xxxx
source

'uranium be most important atomic energy source'

Interlingual operations: NVIQ*R permutes VIQ2 'most important' and NB5 'atomic energy'.

Figure 33:


IND2
AV2
IND
WD
NB5
NRS
VIH4
NB3
NB3*R
VTH2
NN5
NDEJY12*R
NB/MLB
NVTH5/NN
JY12
WD
NVTH/NN
JY12
NB/MLB
NRS
VTH
NN
為
分析
之
便
我們
利用
記錄譜仪
wei
fen-shi
zhi
bian
wo-men
li-yong
ji-lu-pu-yi
for/
for+the+purpose+of
analysis
of
ease/
convenience
we
use
recording+
spectrometer

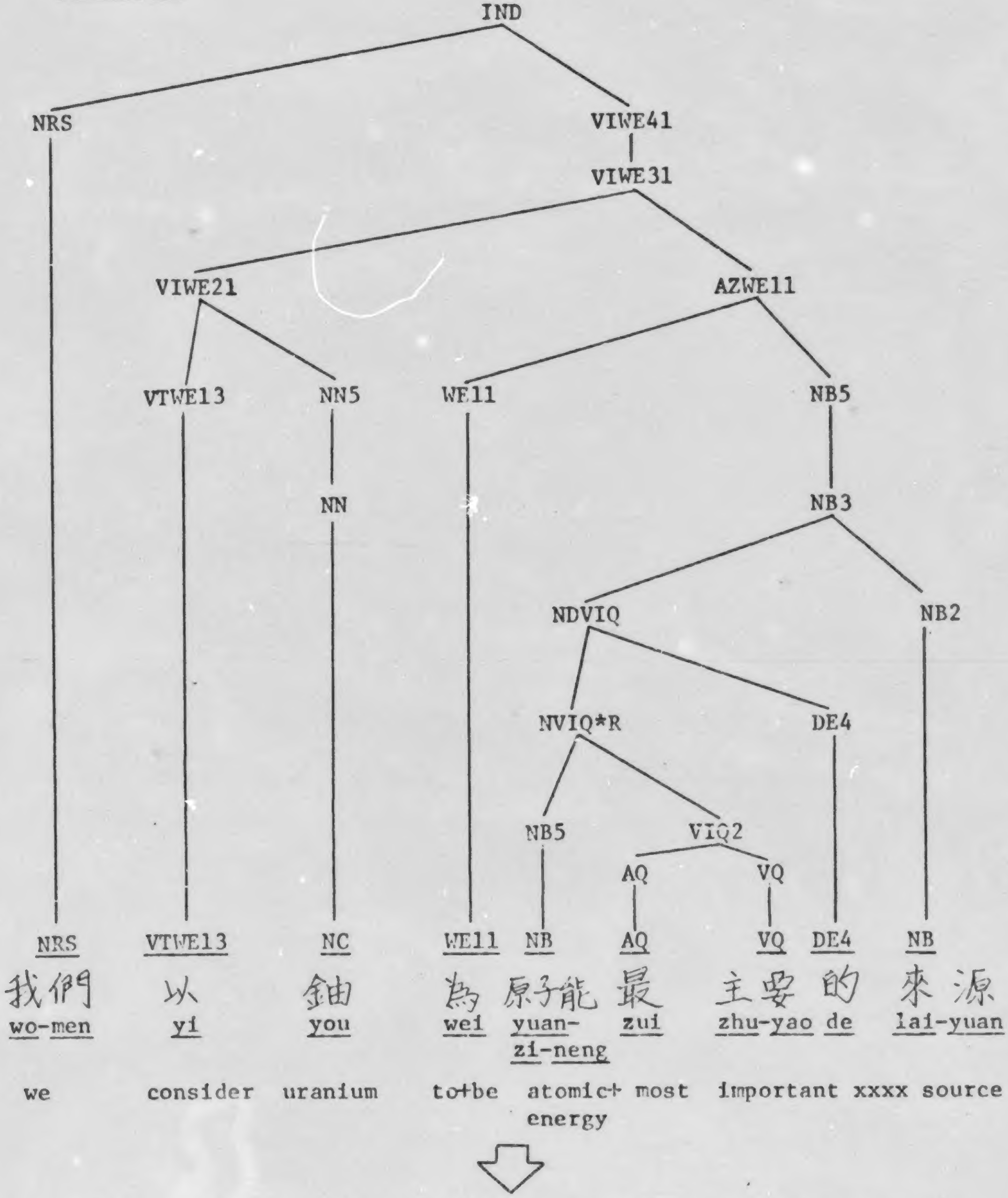
'for/for the purpose of ease/convenience of analysis we use recording spectrometer'

Interlingual operations:

(a) NDEJY*12 permutes NVTH5/NN 'analysis' and JY12 'of'.

(b) NB3*R permutes NDEJY12*R 'of analysis' and NB/MLB 'ease/convenience'.

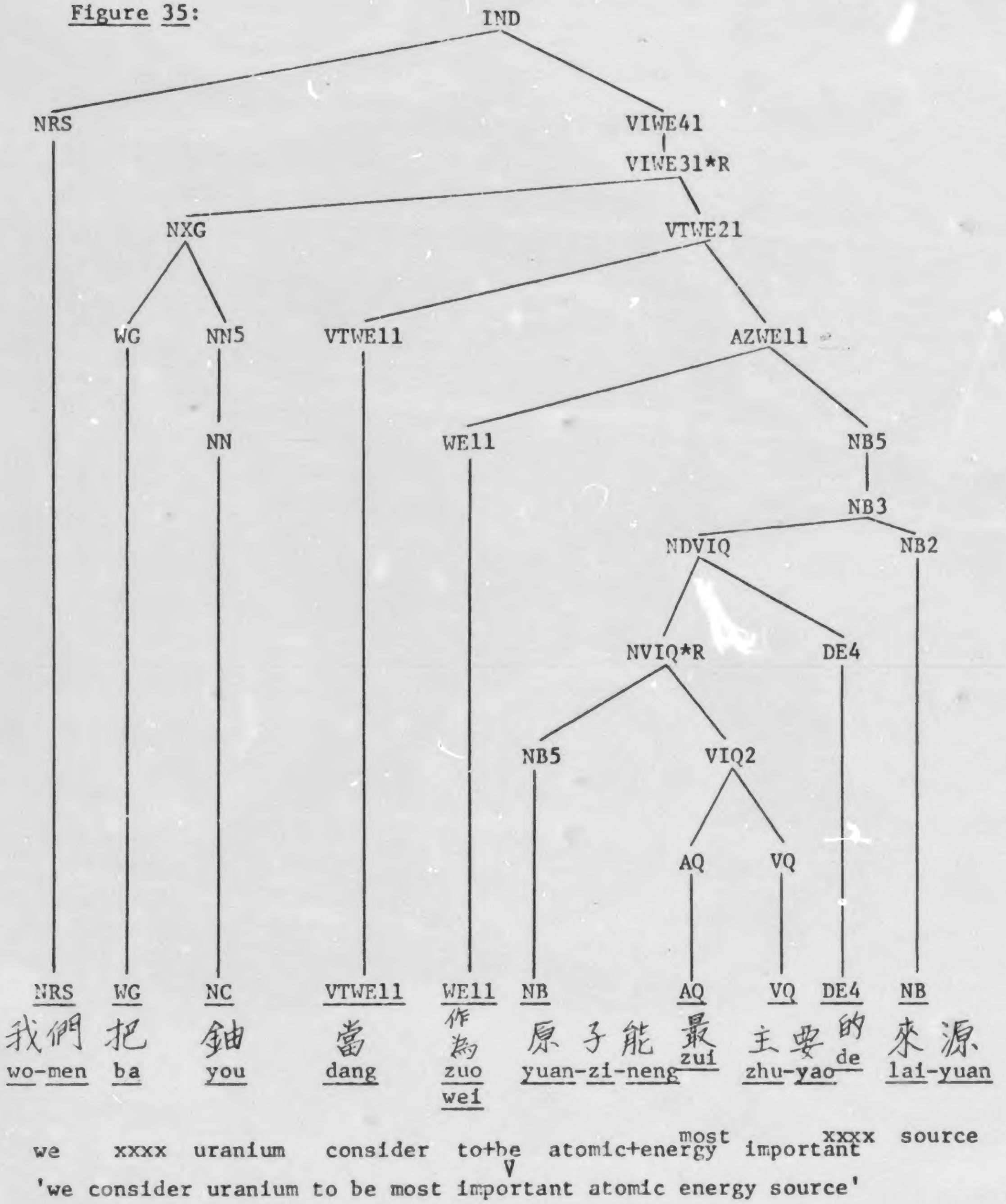
Figure 34:


IND
NRS
VIWE41
VIWE31
VIWE21
AZWE11
VTWE13
NN5
WE11
NB5
NN
NB3
NDVIQ
NB2
NVIQ*R
DE4
NB5
VIQ2
AQ
VQ
NRS
VTWE13
NC
WE11
NB
AQ
VQ
DE4
NB
我們
以
鈾
為
原子能
最
主要
的
來源
wo-men
yi
you
wei
yuan-zi-neng
zui
zhu-yao
de
lai-yuan
we
consider
uranium
to+be
atomic+ energy
most
important
xxxx
source

'we consider uranium to be most important atomic energy source'

Interlingual operations: NVIQ*R permutes NB5 'atomic energy' and VIQ2 'most important'

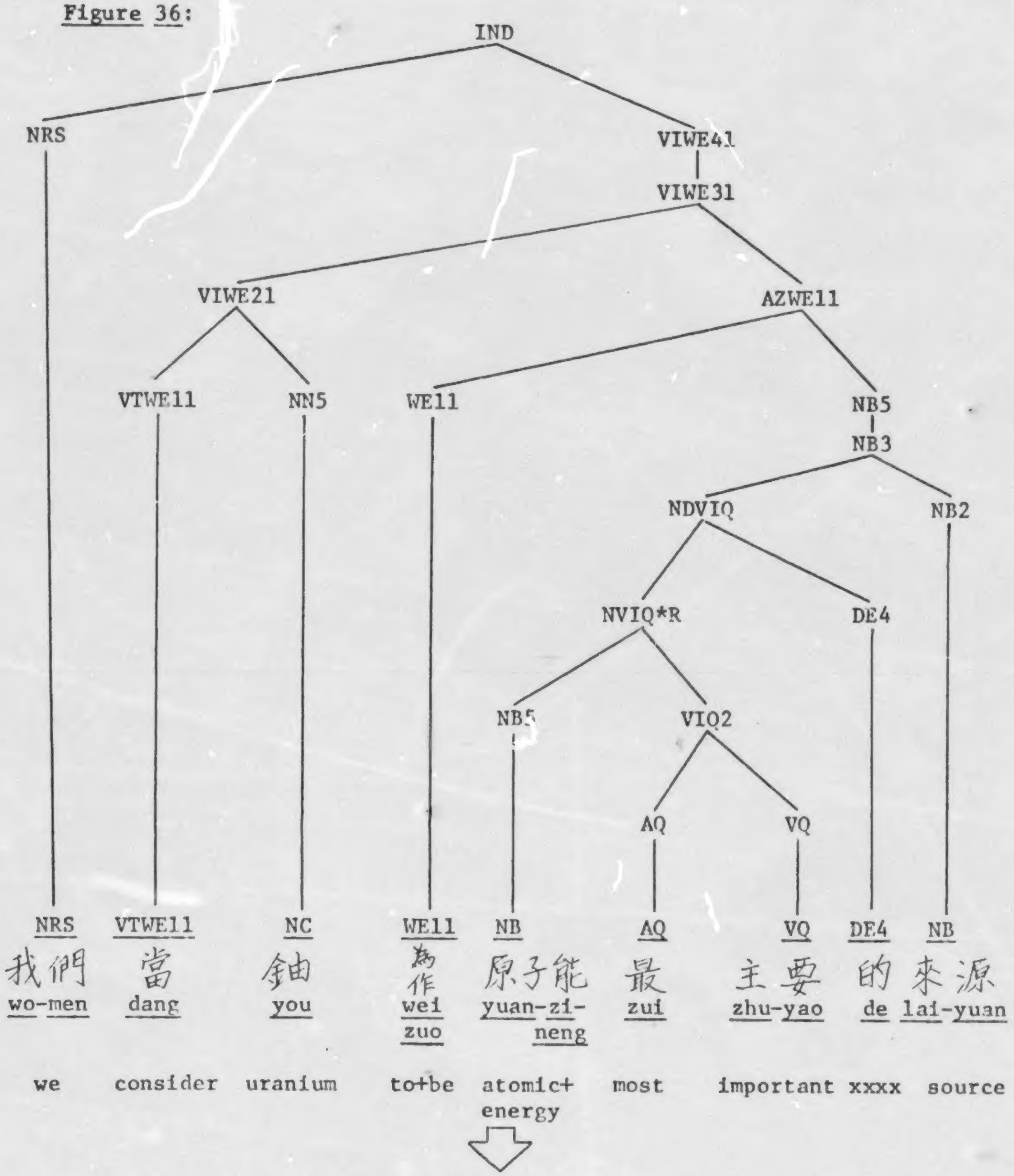
Figure 35:


IND
NRS
VIWE41
VIWE31*R
NXG
VTWE21
WG
NN5
VTWE11
AZWE11
NN
WE11
NB5
NB3
NDVIQ
NB2
NVIQ*R
DE4
NB5
VIQ2
AQ
VQ
NRS
WG
NC
VTWE11
WE11
NB
AQ
VQ
DE4
NB
我們
把
鈾
當
作為
原子能
最
主要
的
來源
wo-men
ba
you
dang
zuo wei
yuan-zi-neng
zui
zhu-yao
de
lai-yuan
we
xxxx
uranium
consider
to+be
atomic+energy
most
important
xxxx
source

'we consider uranium to be most important atomic energy source'

Interlingual operations: (a) NVIQ*R permutes NB5 'atomic energy' with VIQ2 'most important'. (b) VIWE31*R permutes NXG 'uranium' with VTWE21 'consider to be most important atomic energy source'.

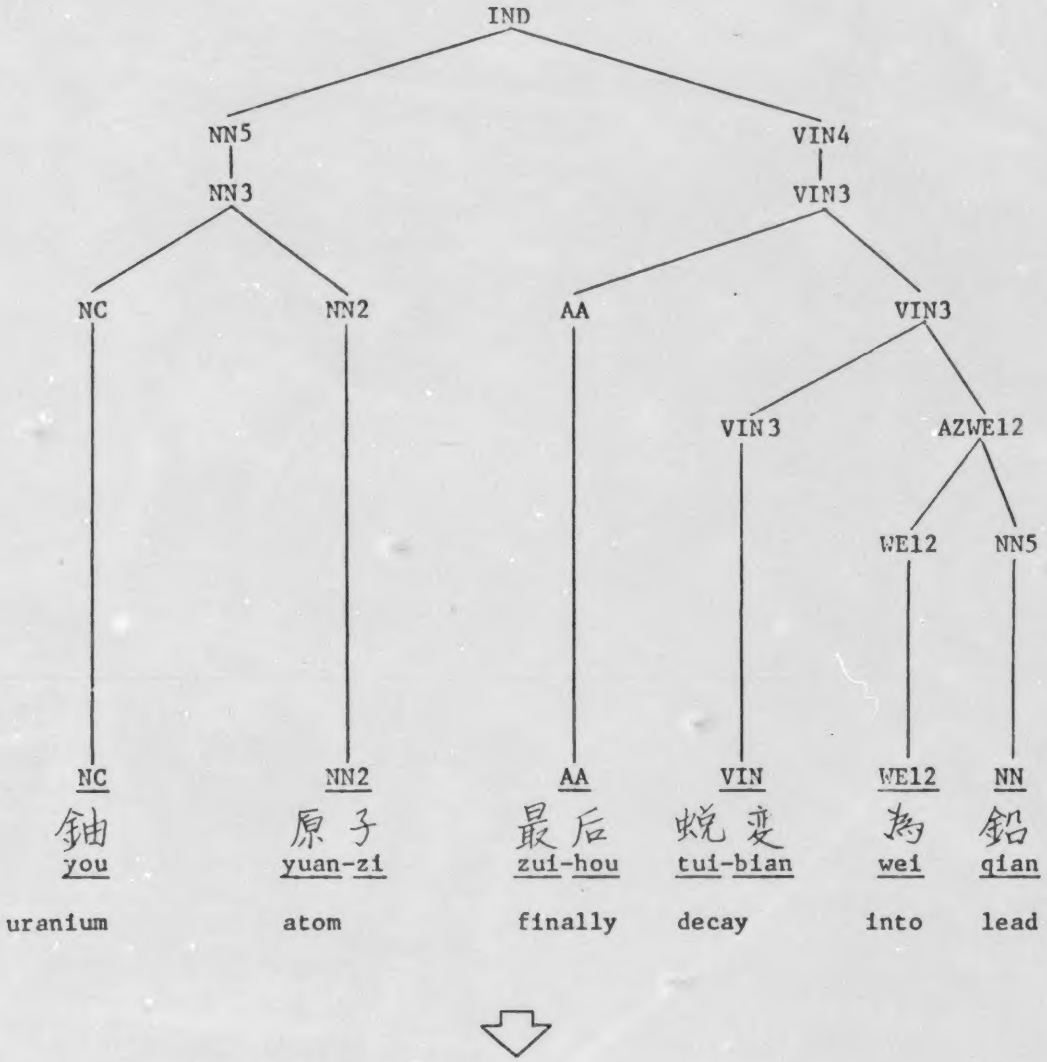
Figure 36:


IND
NRS
VIWE41
VIWE31
VIWE21
AZWE11
VTWE11
NN5
WE11
NB5
NB3
NDVIQ
NB2
NVIQ*R
DE4
NB5
VIQ2
AQ
VQ
NRS
VTWE11
NC
WE11
NB
AQ
VQ
DE4
NB
我們
wo-men
當
dang
鈾
you
為作
wei
zuo
原子能
yuan-zi-
neng
最
zui
主要
zhu-yao
的
de
来源
lai-yuan
we
consider
uranium
to+be
atomic+
energy
most
important
xxxx
source

'we consider uranium to be most important atomic energy source'

Interlingual operations: NVIQ*R permutes NB5 'atomic energy' and VIQ2 'most important'.

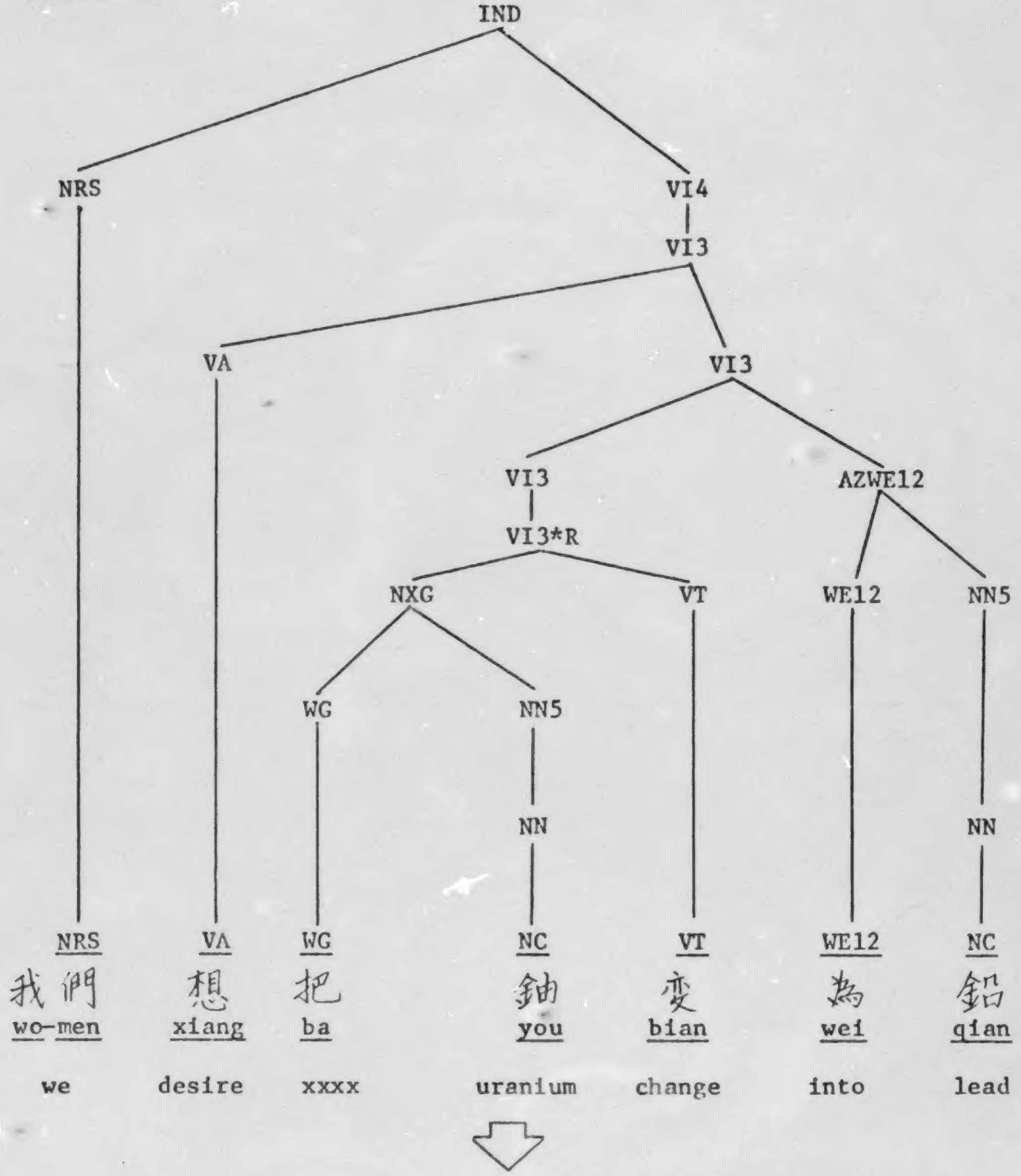
Figure 37:


IND
NN5
NN3
NC
NN2
VIN4
VIN3
AA
VIN3
VIN3
AZWE12
WE12
NN5
NC
NN2
AA
VIN
WE12
NN
鈾
you
uranium
原子
yuan-zi
atom
最后
zui-hou
finally
蜕変
tui-bian
decay
為
wei
into
鉛
qian
lead

'uranium atom finally decay into lead'

Interlingual operations: none

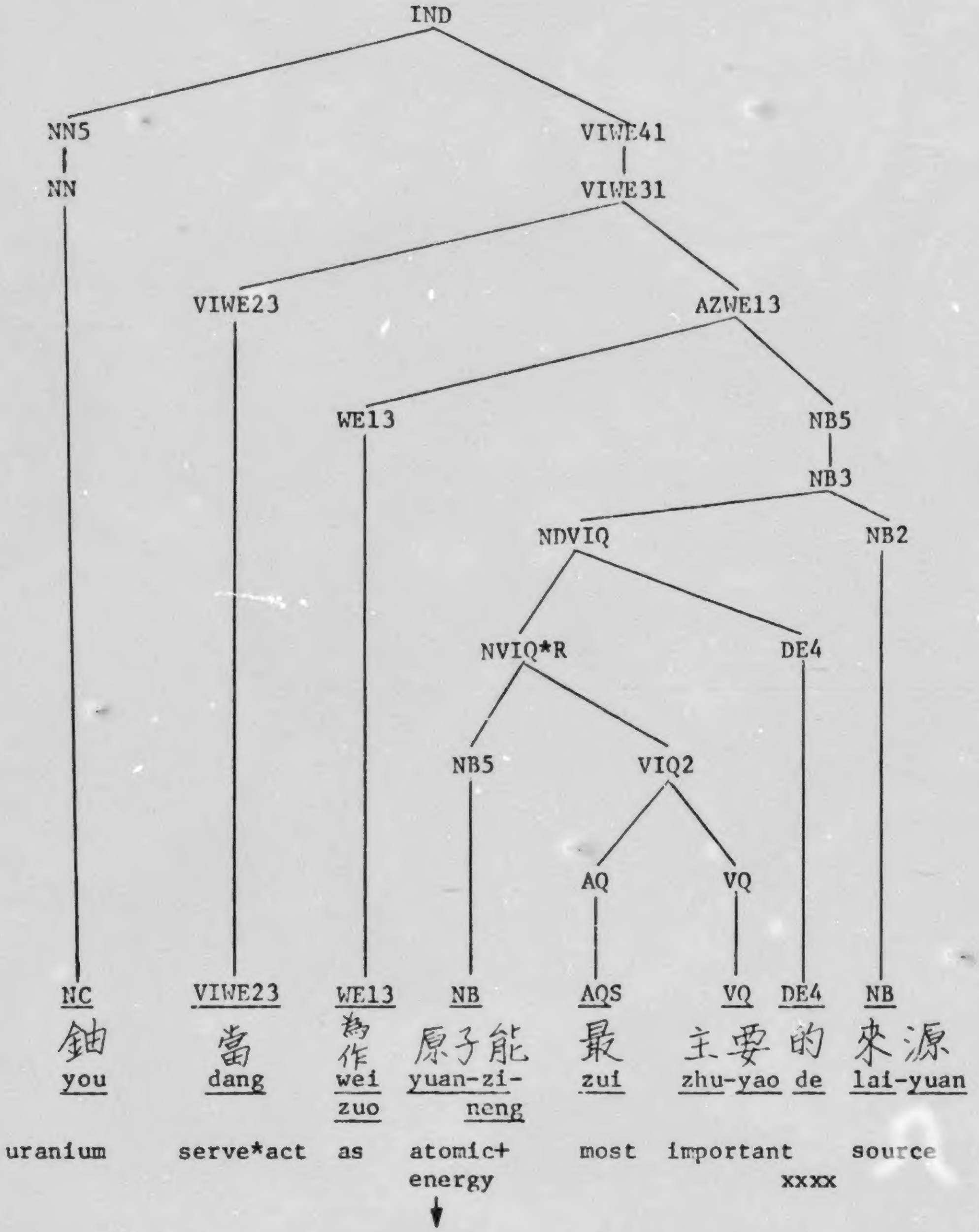
Figure 38:


IND
NRS
VI4
VI3
VA
VI3
VI3
AZWE12
VI3*R
NXG
VT
WE12
NN5
WG
NN5
NN
NN
NRS
VA
WG
NC
VT
WE12
NC
我們
想
把
鈾
变
為
鉛
wo-men
xiang
ba
you
bian
wei
qian
we
desire
xxxx
uranium
change
into
lead

'we desire change uranium into lead'

Interlingual operations: VI3*R permutes NXG 'uranium' and VT 'change'

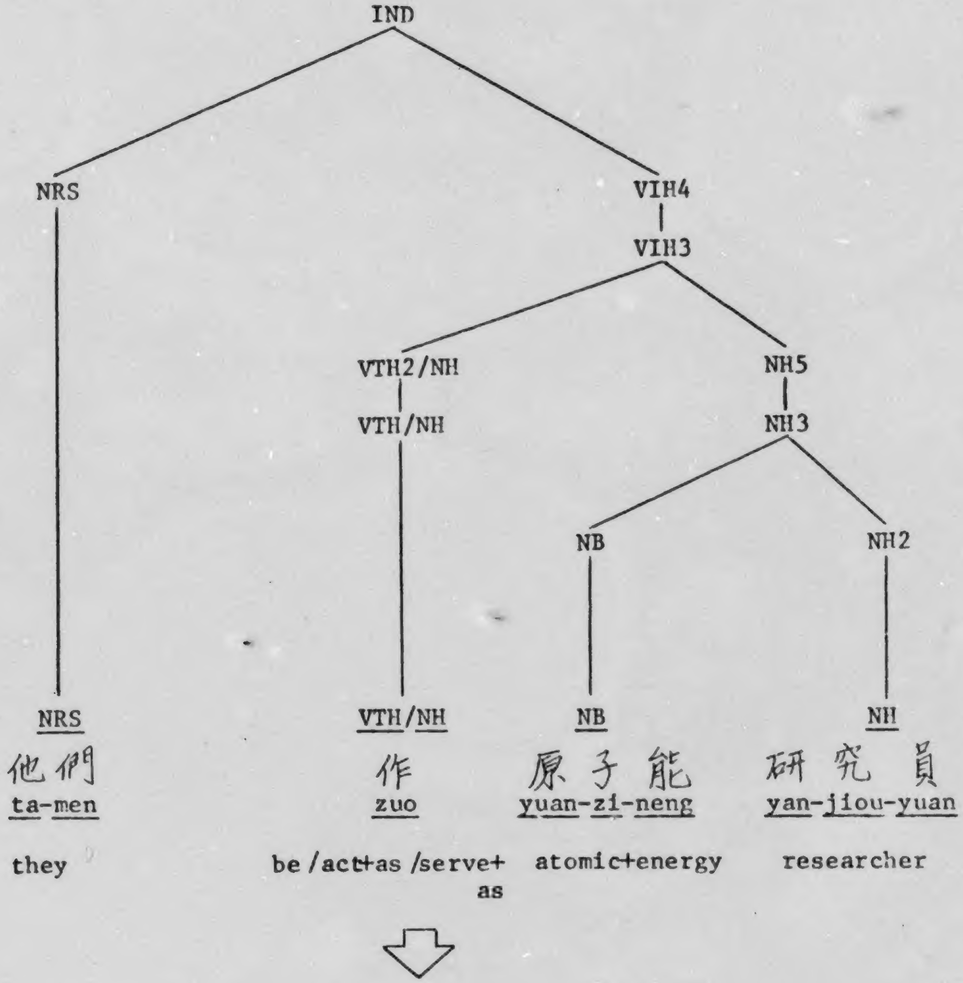
Figure 39:


IND
NN5
NN
VIWE41
VIWE31
VIWE23
AZWE13
WE13
NB5
NB3
NDVIQ
NB2
NVIQ*R
DE4
NB5
VIQ2
AQ
VQ
NC
VIWE23
WE13
NB
AQS
VQ
DE4
NB
鈾
當
為作
原子能
最
主要
的
來源
you
dang
wei zuo
yuan-zi-neng
zui
zhu-yao
de
lai-yuan
uranium
serve*act
as
atomic+ energy
most
important
xxxx
source

'uranium serve*act as most important atomic energy source'

Interlingual operations: NVIQ*R permutes NB5 'atomic energy' and VIQ2 'most important'

Figure 40:


IND
NRS
VIH4
VIH3
VTH2/NH
NH5
VTH/NH
NH3
NB
NH2
NRS
VTH/NH
NB
NH
他們
ta-men
作
zuo
原子能
yuan-zi-neng
研究員
yan-jiou-yuan
they
be /act+as /serve+ as
atomic+energy
researcher

'they be /act as /serve as atomic energy researcher'

Interlingual operations: none

The examples we have just seen have made use of highly differentiated codings of certain key function words, particularly de 的, directional verbs and complements, and yi/wei/zuo 以為作. While these multiple codings are essential for the correct handling of interlingual transfers, they are in part responsible for what we have elsewhere (Wang et al. 1974: 20) termed a "sentence dictionary", namely, "the selection of items from the full lexicon which could be present in the sentence depending on how alternative segmentations and grammatical functions are resolved." This selection is "ordinarily two or three times the number of lexical items which we actually determine later to be present." One other major multiple coding having more to do with content items than function words can be cited: N_x vs. DP_x; i.e. noun vs. derived descriptive adjective, e.g. fang-she-xing 放射性 NB 'radioactivity' and DP 'radioactive', qiu-xing 球形 NB 'sphere' DP 'spherical', ke-xue 科学 NB 'science' DP 'scientific', sheng-li 生理 NB 'physiology' DP 'physiological, dian-zi 电子 NN 'electron' DP 'electronic', wei-li 微粒 NN 'corpuscle' DP 'corpuscular', etc.

After a closer examination of the above material, one might question the justification for including at least certain parts of it in an ostensibly "Chinese" grammar and lexicon. The most obvious cases in point would be those where the English glosses were zero (cf. codes DE4, VD12, VG14, 1VAK-VA, etc.), as well as the N_x - DP_x pairs just mentioned, where Chinese shows no overt morphological marking that could be equated with a verb-noun or noun-adjective derivation.

To be sure, the ideal place to handle many of these cases would be an interlingual component of the grammar separate from the Chinese component. This is indeed our eventual goal. However, it must be pointed out that our present system, which integrates both Chinese and interlingual analyses, accomplishes basically the same thing as a system of strictly separated components. Moreover, that English requires a special derivation for some nominally-related modifiers (cf. radioactive vs. radioactivity) but not others (cf. neutron vs. neutron), or has zero-gloss equivalents for certain Chinese morphemes, are facts about English which must be made available to the system before the final output of the English string: at any rate, such facts cannot be reduced or eliminated by shuttling them back and forth from one component to another. The "English" information contained in the glosses of the item types just described will thus be quite valuable once a true separation of Chinese and interlingual components is effected.

Even more importantly, we feel that our differential codings are substantially justified on grounds internal to Chinese syntax and semantics. While we have undertaken parsing of dictionary entries as an aid to lexicographic and grammatical research, we have been careful to keep in mind that such research has its practical limits. We do not intend to claim, as so many scholars mistakenly have, that because the Chinese writing system does not delineate lexical items, the Chinese language therefore has no lexical items, or else each character is equivalent to a lexical item, to be freely compounded with any character adjacent to it. A few examples from our texts will suffice to illustrate this point.

(1) fan-ying 反应 'to react' - Graphically speaking, this lexical item is composed of the characters fan 反 and ying 应. By itself, 'fan' 反 means 'return', 'turn over', while 'ying' 应 can mean 'respond', 'reply', 'assent'. If we were to examine the contexts in which fan-ying 反应 occurred in Chinese, however, it would be obvious that the meaning that we wished to assign to it could hardly be equated with the sum of the meanings of graphic components, any more than the meaning of 'react' in English is unequivocally derivable from 're' + 'act'. We therefore conclude that fan-ying 反应 is a lexical item, and that it must be listed as a separate entry in a Chinese dictionary.

(2) yuan-zi-he 原子核 'atomic nucleus'

Again, this item consists of several graphic components:

yuan	原	'original', 'source'
zi	子	noun suffix
he	核	'pit (of fruit), nucleus'

The derivation of yuan-zi 原子 'atom' from 'yuan' + 'zi' poses the same types of problems as in example (1). Moreover, the fact that 'he' 核 here is to be interpreted as 'nucleus' is just as much a statement about Chinese word-formation as it is about the desired English translation: That is, yuan-zi-he 原子核 is structurally no different from tao-zi-he 桃子核 'peach pit', where the interpretation of 'he' 核 is quite another matter. It is only on a much deeper, less accesible semantic level that one can talk of 'he' 核 as a morpheme with a single connotation. In an important sense, then, 原子核 yuan-zi-he , as well as 桃子核 tao-zi-he must be considered lexical items in Chinese.

Similar types of arguments hold for form classes in Chinese (i.e. nouns, verbs, adverbs, etc.). As mentioned in Wang et al. 1974:19, it is true that Chinese lexical items in context are typically not directly marked as to tense, number, case, etc. However, even if we were to apply more diffuse contextual criteria, it would soon become evident that (a) certain groups of lexical items were quite restricted as to the types of syntactic roles they could fill, or (b) that what graphically appeared to be a single lexical item actually filled several distinct syntactic roles relatable only on a highly abstract level. As concerns type (a), it is easy to find lexical items whose functions are limited to one or two form classes. The following list of examples is by no means exhaustive:

Nouns only (i.e. cannot be used as verbs, but may be subjoined to other nouns as de facto modifiers):

原子 yuan-zi 'atom', 核子 he-zi 'nucleon', 方法 fang-fa 'method', 結构 jie-gou 'structure', 加速器 jia-su-qi 'accelerator', 理論 li-lun 'theory', 世界 shi-jie 'world', 燃料 ran-liao 'fuel'.

Verbs only: 应該 ing-gai 'must, 想 xiang 'think', 使 shi 'cause', 看見 kan-jian 'see', 覺得 jue-de 'feel', 離開 li-kai 'leave', 引起 yin-qi 'cause', 稳定 wen-ding 'be stable'.

Conjunctions or adverbs only: 已經 yi-jing 'already', 非常 fei-chang 'unusually', 因为 in-wei 'because', 所以 suo-yi 'therefore', 雖然 sui-ran 'although', 要不然 yao-bu-ran 'otherwise', 馬上 ma-shang 'immediately', 常常 chang-chang 'often', 如果 ru-guo 'if'.

Type (b) cases (see Table 2) are somewhat more complex: all words which now function as (1) prepositions, directional complements, conjunctions, and other function words are graphically identical with items which were once, or still are, (2) nouns and verbs. In some cases, the semantic relationship between (1) and (2) is fairly obvious; in others, no true relationship exists: a character from (2) was *borrowed* (usually on the basis of homophony) to represent an item in (1) (cf. *de* 的 in Table 2).

At first glance, type (b) cases might seem amenable to uniform treatment as nouns and/or verbs with the other syntactic functions being secondarily derived. Such an approach would immediately run into difficulties: while there may be a relationship between the noun/verb and other functions, it is different for each item; there is no *principled* way of predicting the derived glosses for Table 2 items as a whole, cf. especially items (1,4,7,11,12,13,14,24,26). But there are even more basic reasons why a unified approach would be impractical. It is true that almost all senses of all items in Table 2 are represented in our texts; however, the environments in which each sense occurs are restricted in very specific ways. For example, all parenthesized glosses represent senses that are appropriate only in Classical Chinese contexts. As has been pointed out elsewhere, Chinese scientific writing not only contains elements of modern spoken style, but also shows a heavy admixture of classical writing style that dates back to before the time of Confucius and has survived only as a literary language. This means that items with parenthesized glosses as nouns and verbs are "bound" morphemes: they must either occur in conjunction with certain classical particles

Table 2

		Verb	Noun	Preposition	Conjunction	Other
(1)	an 桉	to press, to comment editorially		according to		
(2)	ba 把	(to take)	(handle)			executive marker; classifier for certain nouns
(3)	bei 被	(to suffer)	(quilt, cover)			passive marker
(4)	bi 比	compare	(comparison, ratio)			comparative marker: 'than'
(5)	bian 便		(convenience)			(adv.) then
(6)	cong 從	(to follow)		from		
(7)	di 的		(target)			
	di	(to be true, accurate)				
	de					subordinative marker
(8)	dui 对	(to face), to be correct	(couplet)	towards, regarding		pair (noun classifier)
(9)	gei 給	to give		to, for		passive marker
(10)	gen 跟	(to follow)		with (comitative)	and (noun conjunction only)	
(11)	he 和	(to mix, blend; be gentle, harmonious)	(sum)	with (comitative)	and (noun conjunction only)	
(12)	zhao 照	to shine, illuminate		according to		
(13)	ji 极		(pole)			(adv.) very, extremely

Table 2 (continued)

	Verb	Noun	Preposition	Conjunction	Other
(14) jiu 就	(to accomplish, realize, approach, condescend)		according to		(adv.) then, soon, at once
(15) kao 靠	to rely on, depend on, lean on		near, along, toward		
(16) lai 来	to come			in order to	(adv.) for the past (+ time measure)
(17) neng 能		(energy, power)			auxiliary verb 'be able'
(18) tong 同	to be the same		with (comitative)	and (noun conjunction only)	
(19) wei 爲	(to be, act as)		for the sake of, as	in order to	passive marker
(20) yan 沿	(to go along, skirt)		along		
(21) yi 以	(to take, consider)		with (instrumental), as	in order to	auxiliary verb formant
(22) yong 用	to use	(use)	with (instrumental)		
(23) you 由		(reason)	from		
(24) yu 于	(to be at)		at		
(25) yu 與	(give)		to, for, with (comitative)	and (noun and verb conjunction)	
(26) zai 在	to be at		at		action progressive marker
(27) ze 則		(rule)			(adv.) then
(28) zi 子		(son, seed)			noun suffix

(e.g. zhi 之 , the classical equivalent of de 的 or else occur in lexical compounds, which are then acceptable in the spoken language as "free" items, or "words". In this latter sense, bound morphemes in Chinese are functionally similar to certain English word-roots derived from Latin, e.g. -duce, which has the general meaning "to draw, pull", but which appears only in combined form in such words as reduce, deduce, induce, produce, transduce, etc. Bound items in Chinese such as those just mentioned are given special codings and are parsed with highly restricted rules that reflect their syntactic and semantic peculiarities; otherwise, they have no independent status as lexical items. The following set illustrates typical occurrences of selected bound items from Table 2 in their noun/verb senses, alone or in compounds:

Table 3

(1) ba 把

In lexical compounds:

火把 huo-ba 'torch'
手把 shou-ba 'handle (n.)'
把手 ba-shou 'railing
把子 ba-zi 'handle (noun)'
把舵 ba-duo 'steer'
把握 ba-wo 'firm hold, assurance'

(2) bei 被

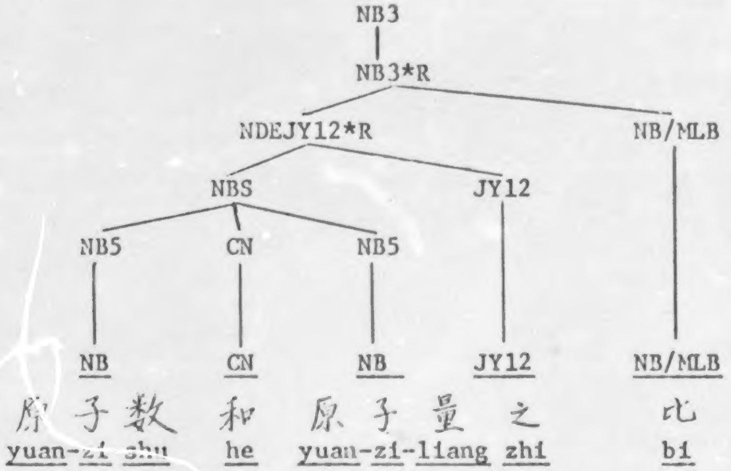
In lexical compounds

單被 dan-bei '(single) sheet'
表被 biao-bei 'coating'
地面植被 di-mian-zhi-bei 'ground cover'
被动 bei-dong 'passive'

Table 3 (continued)

(3) bi 比

(a) Alone: NB/MLB 'ratio/comparison' (Figure 41)



NB3
NB3*R
NDEJY12*R
NB/MLB
NBS
JY12
NB5
CN
NB5
NB
CN
NB
JY12
NB/MLB
原子数 和 原子量 之 比
yuan-zi shu he yuan-zi-liang zhi bi

atomic+number and atomic+weight of ratio/comparison

→ 'ratio/comparison of atomic number and atomic weight'

Interlingual operations:

NDEJY12*R permutes NBS 'atomic number and atomic weight' and JY12 'of'.

NB3*R permutes NDEJY12*R 'of atomic number and atomic weight' and NB/MLB 'ratio/comparison'.

(b) In lexical compounds:

比例 bi-li NB 'ratio'

比能 bi-neng NB 'specific energy'

比拟 bi-ni NB 'analog'

比热 bi-re NB 'specific heat'

比較 bi-jiao VTH 'compare' AQ 'comparatively'

差比 cha-bi NB 'increment ratio'

对比 dui-bi VTH 'contrast/compare'

同比 tong-bi NB 'common denominator'

正比 zheng-bi NB 'direct proportion'

Table 3 (continued)

(4) bian 便

(a) Alone: NB/MLB 'ease/convenience' (see Figure 33).

(b) In lexical compounds

方便 fang-bian VQ 'convenient'

輕便 qing-bian VQ 'portable'

以便 yi-bian CV 'so+as+to+facilitate'

(5) di 的

In lexical compounds:

标的 biao-di NN 'target'

的确 di-que VQ 'accurate'

目的 mu-di NB 'goal'

目的地 mu-di-di NP 'destination'

中的 zhong-di VIH2 'hit target'

(6) dui 对

In lexical compounds:

对比 dui-bi VTH 'contrast/compare'

对付 dui-fu VTH 'deal with'

对角 dui-jiao DP 'diagonal'

对数 dui-shu NB 'logarithm'

绝对 jue-dui VQ 'absolute'

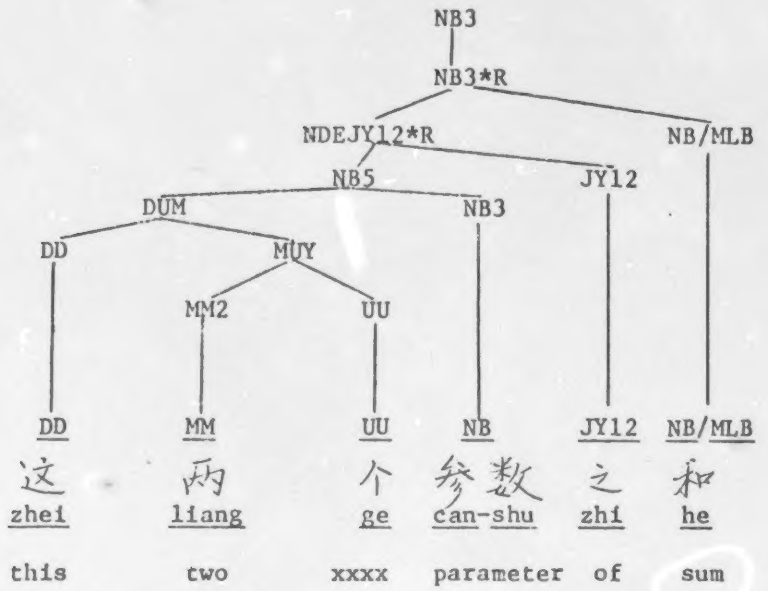
相对 xiang-dui VQ 'relative'

反对 fan-dui VT 'oppose'

离子对 li-zi-dui NN 'ion-pair'

(7) he 和 Table 3 (continued)

(a) Alone: NB/MLB 'sum'(Figure 42)


NB3
NB3*R
NDEJY12*R
NB/MLB
NB5
JY12
DUM
NB3
DD
MUY
MM2
UU
DD
MM
UU
NB
JY12
NB/MLB
这
两
个
参数
之
和
zhei
liang
ge
can-shu
zhi
he
this
two
xxxx
parameter
of
sum

'sum of this two parameter'

Interlingual operations:

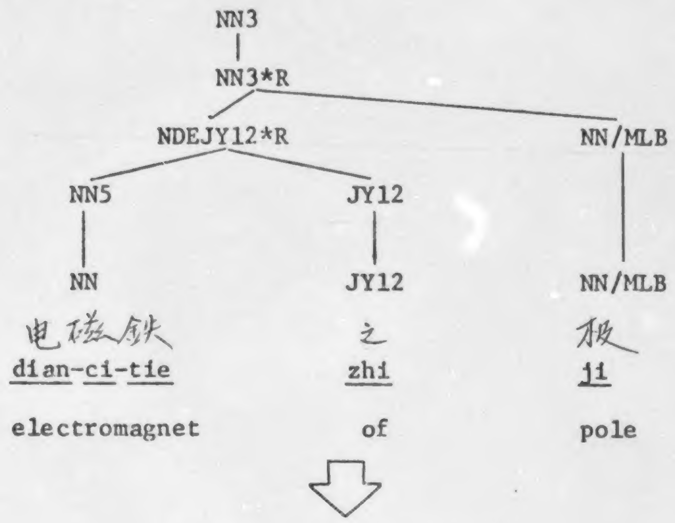
(a) NDEJY12*R permutes NB5 'this two parameter' and JY12 'of'.
(b) NB3*R permutes NDEJY12*R 'of this two parameter' and NB/MLB 'sum'.

(b) In lexical compounds:

和数	he-shu	NB	'sum'
和頻率	he-pin-lu	NB	'sum+frequency'
和平	he-ping	NB	'peace'
和諧	he-xie	NB	'harmony'
調和	diao-he	DP	'harmonic'
調和	tiao-he	VT	'regulate'
總和	zong-he	NB	'sum'

(8) ji 极 Table 3 (continued)

(a) Alone: NN/MLB 'pole' (Figure 43)


NN3
NN3*R
NDEJY12*R
NN/MLB
NN5
JY12
NN
JY12
NN/MLB
电磁铁
之
极
dian-ci-tie
zhi
ji
electromagnet
of
pole

'pole of electromagnet'

Interlingual operations:

NDEJY12 permutes NN5 'electromagnet' and JY12 'of'.

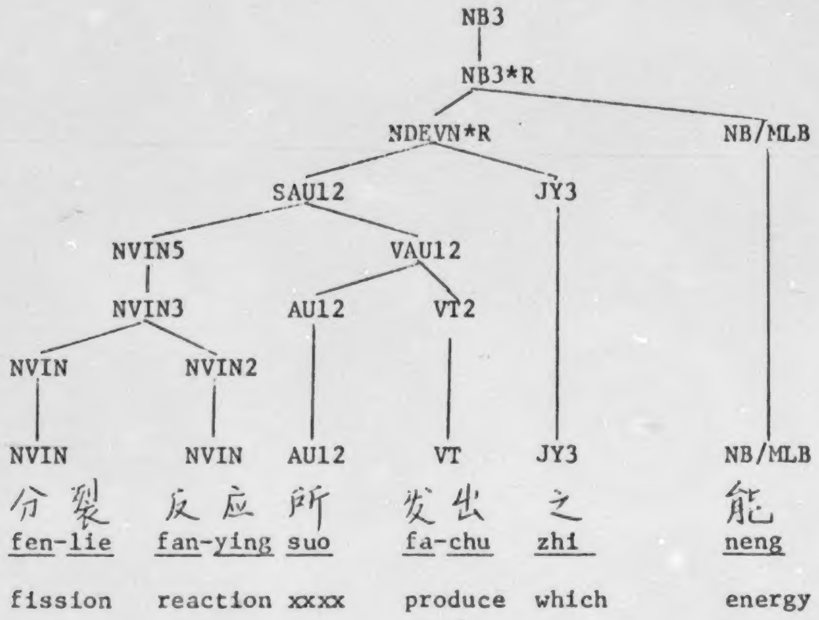
NN3*R permutes NDEJY12*R 'of electromagnet' and NN/MLB 'pole'.

(b) In lexical compounds:

极性 ji-xing NB 'polarity'
极化 ji-hua VTN 'polarize'
北极 bei-ji NP 'North Pole'
电极 dian-ji NN 'electrode'
环极 huan-ji DP 'circumpolar'
陰极 yin-ji NN 'cathode'
陽极 yang-ji NN 'anode'

(9) neng 能

(a) Alone: NB/MLB 'power/energy' (Figure 44)


NB3
NB3*R
NDEVN*R
NB/MLB
SAU12
JY3
NVIN5
VAU12
NVIN3
AU12
VT2
NVIN
NVIN2
NVIN
NVIN
AU12
VT
JY3
NB/MLB
分裂
反应
所
发出
之
能
fen-lie
fan-ying
suo
fa-chu
zhi
neng
fission
reaction
xxxx
produce
which
energy

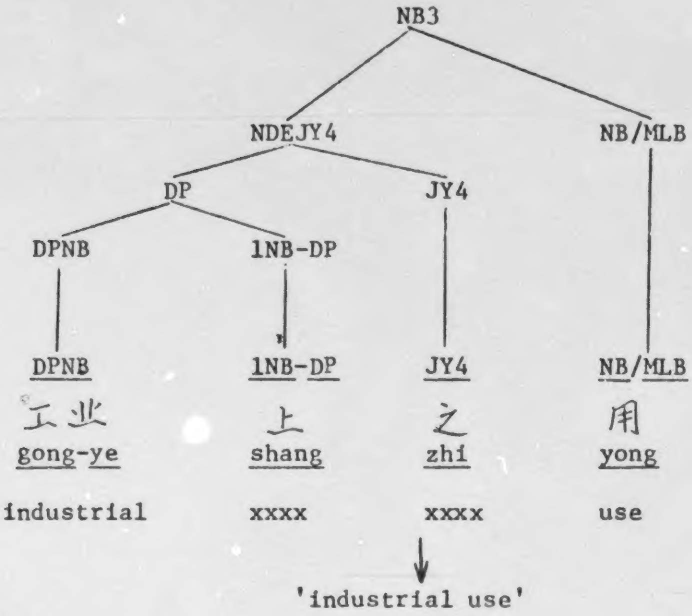
'energy which fission reaction produce'

(b) In lexical compounds:

能量	neng-liang	NB	'amount+of+energy'
能譜	neng-pu	NB	'energy+spectrum'
能态	neng-tai	NB	'energy+state'
能源	neng-yuan	NB	'energy+source'
动能	dong-neng	NB	'kinetic+energy'
热能	re-neng	NB	'thermal+energy'
位能	wei-neng	NB	'potential+energy'
原子能	yuan-zi-neng	NB	'atomic+energy'

(10) yong 用 Table 3 (continued)

(a) Alone: NB/MLB 'use' (Figure 45)


NB3
NDEJY4
NB/MLB
DP
JY4
DPNB
1NB-DP
DPNB
1NB-DP
JY4
NB/MLB
工业
gong-ye
industrial
上
shang
xxxx
之
zhi
xxxx
用
yong
use
'industrial use'

(Note that yong as a noun is obligatorily bound; however, it can be used freely as a verb, cf. Figure 31.)

(b) In lexical compounds:

用处	yong-chu	NB	'usefulness'
用量	yong-liang	NB	'amount+used'
用途	yong-tu	NB	'application'
用具	yong-ju	NN	'utensil'
使用	shi-yong	VTH	'use'
有用	you-yong	VQ	'useful'
应用	ying-yong	VTH	'apply'
作用	zuo-yong	NB	'function'

As a matter of fact, when the items above occur in isolation, they do so far more frequently in their derived prepositional, conjunctive, and other senses than in the noun/verb senses just exemplified. Even if we eliminate 的 *di* 'target' vs. 的 *de* (subordinative particle) as a case of homography rather than true semantic relationship, there still remain a wide variety of examples which illustrate our point:

Table 4

(1) *ba* 把

(a) UU (count-noun classifier) 'xxxx', e.g. 三 把 降落伞 *san ba jiang-luo-san* 'three parachutes'.

(b) UM (mass-noun classifier) 'bundle/handful', e.g. 一把沙 *i ba sha* 'a handful of sand'.

(c) WG (pre-transitive topic marker) 'xxxx' (see *Figures 20, 26, 35, 38*; also cf. Chao 1968: 342-50, Cheung 1973, Frei 1956, Teng 1972, Thompson 1973).

(2) *bi* 比 WQ 'than' (participates in comparative structures; see *Figures 11-12*, also cf. Wang et. al. 1974:74-77).

(3) *bian* 便 AA 'then' (post-subject adverb; see *Figures* 46-47). The adverbs 就 *jiu* and 则 *ze* (Table 2) can replace *bian* in this context.

(4) *he* 和 CN 'and' (noun conjunction; see *Figure 41*). The conjunctions 跟 *gen*, 同 *tong*, 与 *yu* (Table 2) can replace *he* here.

(5) *ji* 极 AQ 'extremely' (degree adverb, can substitute for AQ *bi-jiao* in *Figure 14*, with accompanying change in meaning).

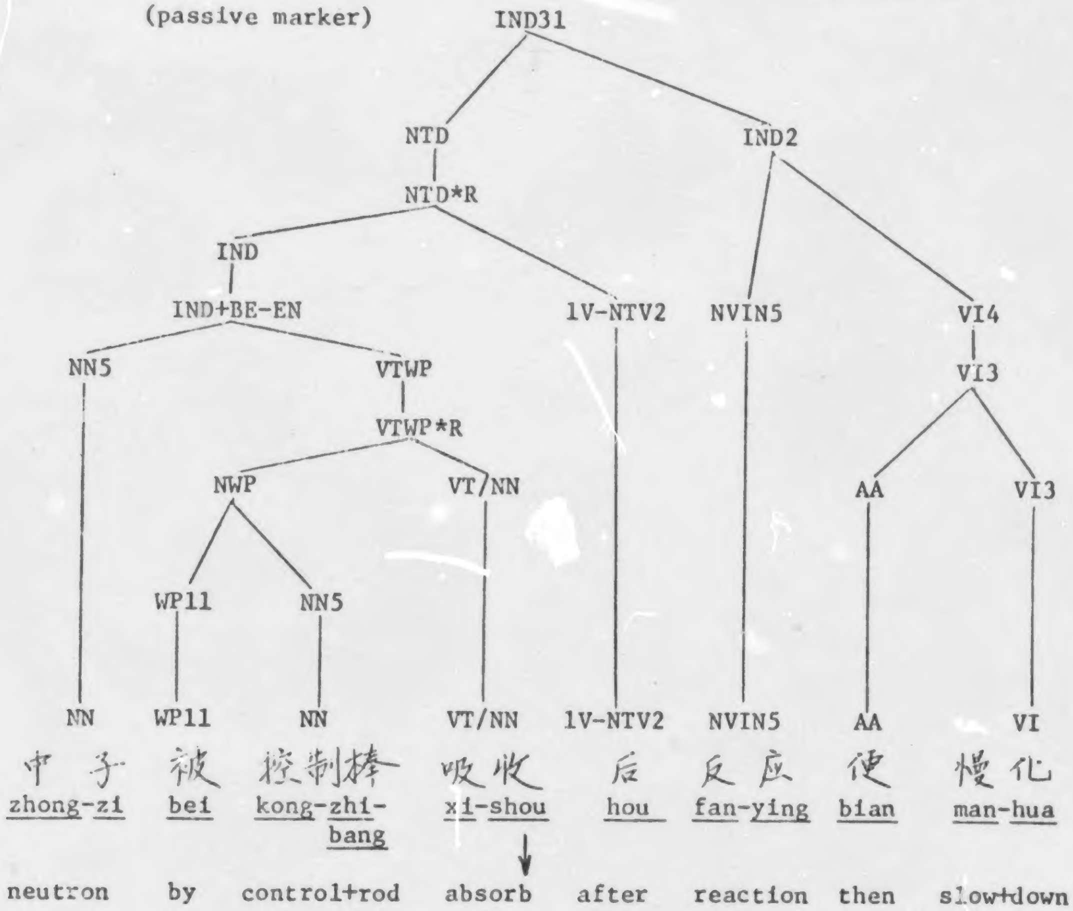
(6) *neng* 能 VA 'be+able" (auxiliary verb; can substitute for *neng-gou* VA 'be+able' in *Figure 27*).

(7) *yong* 用 WZ 'with' (instrumental preposition; see *Figure 26*).

Table 4 (continued)

(8a) 被 bei WP11 'by'
(passive marker)

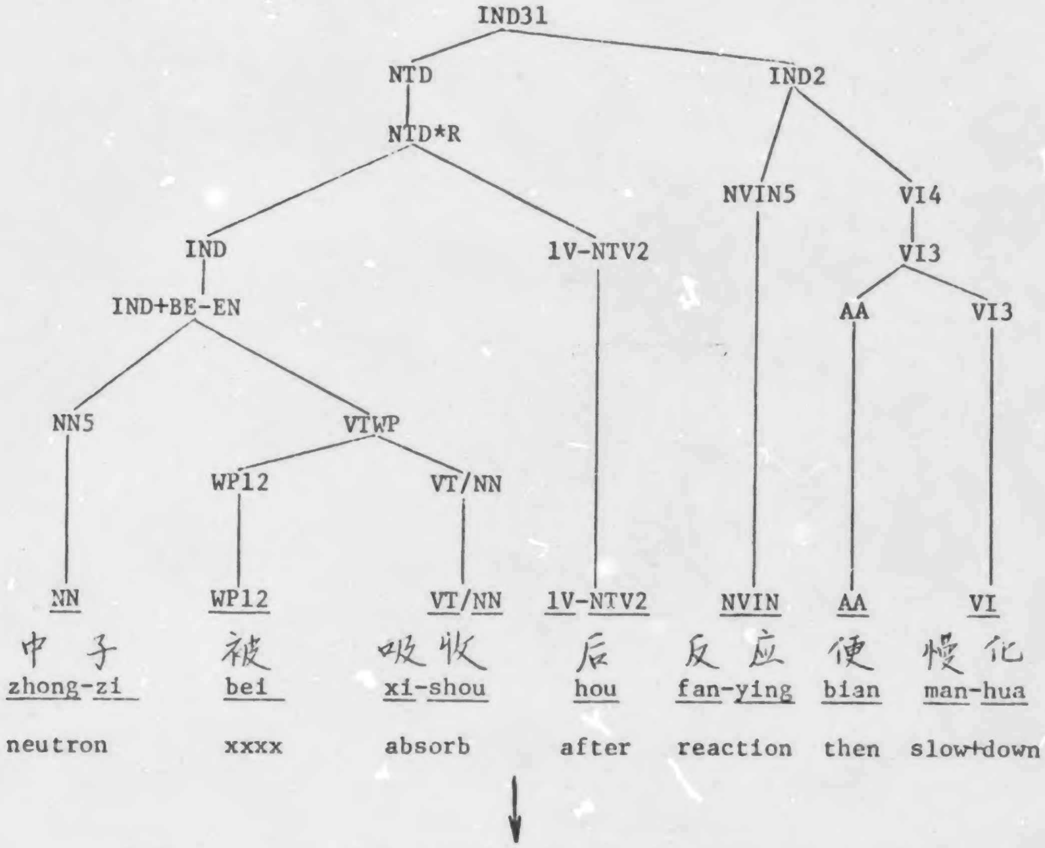
(Figure 46)


IND31
NTD
IND2
NTD*R
IND
IND+BE-EN
1V-NTV2
NVIN5
VI4
NN5
VTWP
VI3
VTWP*R
NWP
VT/NN
AA
VI3
WP11
NN5
NN
WP11
NN
VT/NN
1V-NTV2
NVIN5
AA
VI
中 子
被
控制棒
吸收
后
反应
便
慢化
zhong-zi
bei
kong-zhi-bang
xi-shou
hou
fan-ying
bian
man-hua
neutron
by
control+rod
absorb
after
reaction
then
slow+down

'after neutron (be) absorb(ed) by control rod reaction then slow down'

Interlingual operations: (a) VTWP*R permutes NWP 'by control rod' and VT/NN 'absorb'.
(b) NTD*R permutes IND 'neutron (be) absorb(ed)by control rod' and 1V-NTV2 'after'.

(8b)被bei WP12 'xxxx' Table 4 (continued) (Figure 47)
(passive marker)


IND31
NTD
IND2
NTD*R
NVIN5
VI4
IND
1V-NTV2
VI3
IND+BE-EN
AA
VI3
NN5
VTWP
WP12
VT/NN
NN
WP12
VT/NN
1V-NTV2
NVIN
AA
VI
中子
被
吸收
后
反应
便
慢化
zhong-zi
bei
xi-shou
hou
fan-ying
bian
man-hua
neutron
xxxx
absorb
after
reaction
then
slow+down

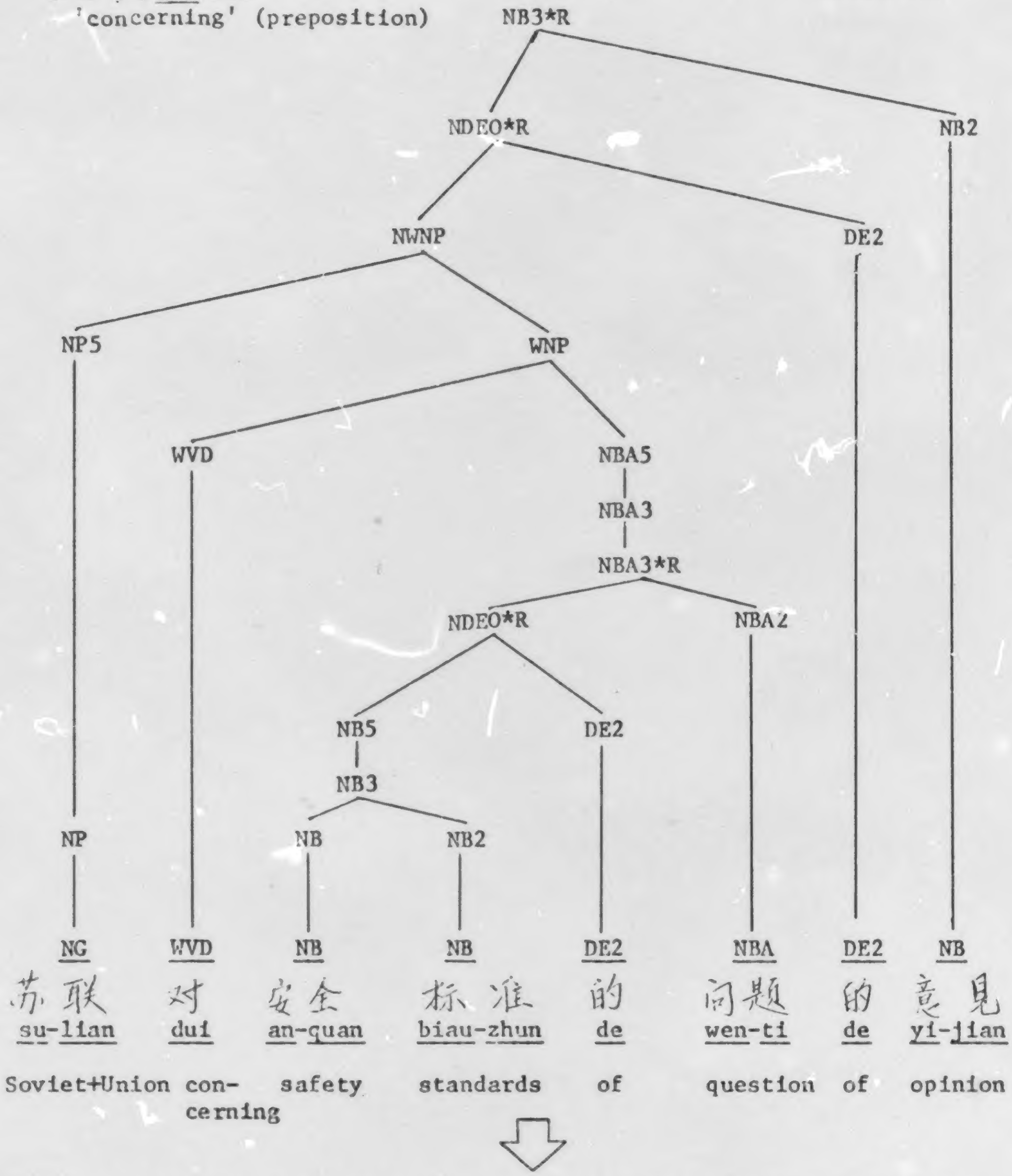
'after neutron (be) absorb(ed) reaction then slow down'

Interlingual operations:

NTD*R permutes IND 'neutron (be) absorb(ed)' with 1V-NTV2 'after'.

Table 4 (continued)

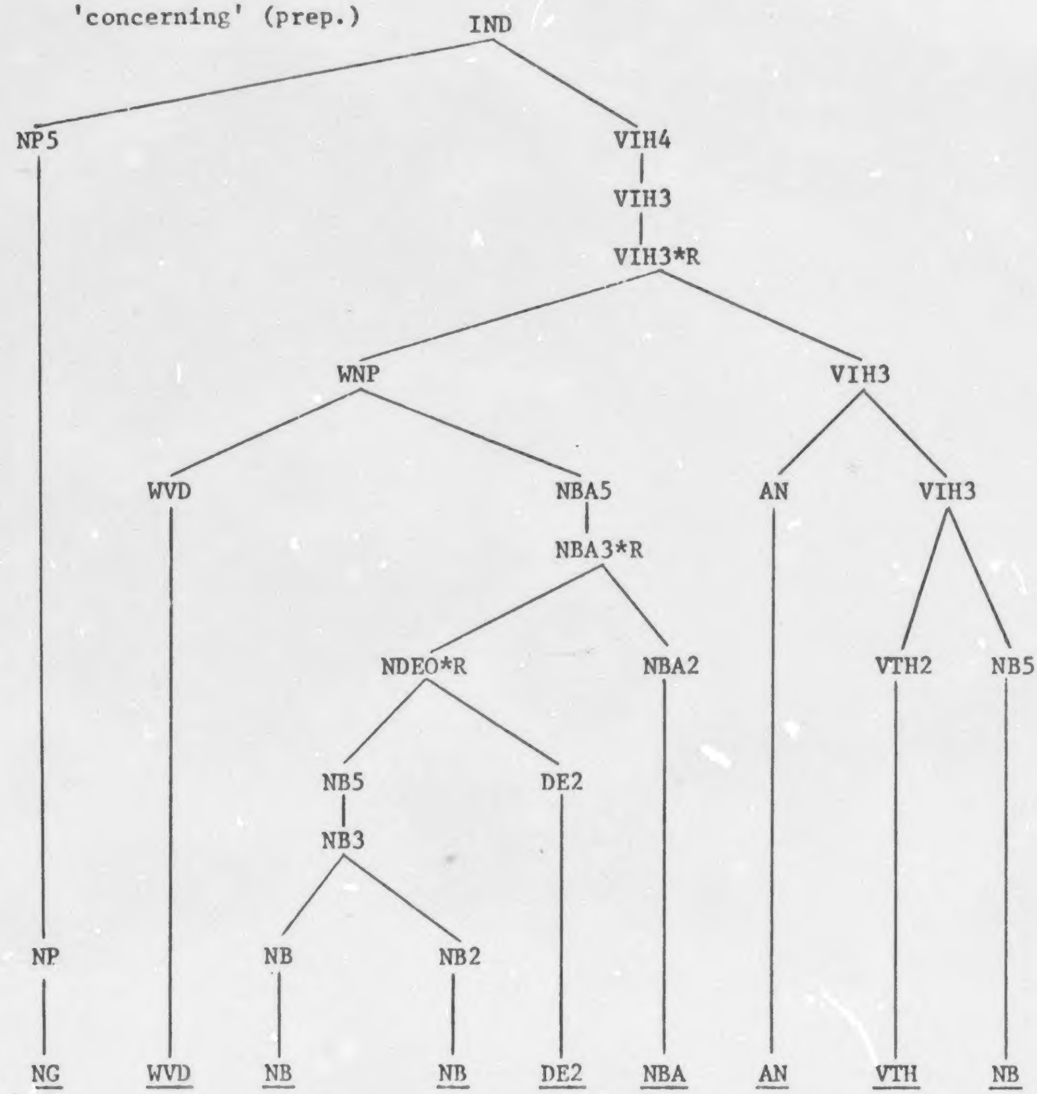
(9a) 对dui WVD (Figure 48)
'concerning' (preposition)


NB3*R
NDEO*R
NB2
NWNP
DE2
NP5
WNP
WVD
NBA5
NBA3
NBA3*R
NDEO*R
NBA2
NB5
DE2
NB3
NP
NB
NB2
NG
WVD
NB
NB
DE2
NBA
DE2
NB
苏联
对
安全
标准
的
问题
的
意見
su-lian
dui
an-quan
biau-zhun
de
wen-ti
de
yi-jian
Soviet+Union
con-cerning
safety
standards
of
question
of
opinion

'opinion of Soviet Union concerning question of safety standards'

Table 4 (continued)

(9b) 对 dui WVD (Figure 49)
'concerning' (prep.)


IND
NP5
VIH4
VIH3
VIH3*R
WNP
VIH3
WVD
NBA5
AN
VIH3
NBA3*R
NDEO*R
NBA2
VTH2
NB5
NB5
DE2
NB3
NP
NB
NB2
NG
WVD
NB
NB
DE2
NBA
AN
VTH
NB

苏联 对 安全 标准 的 问题 没有 发表 意見

su-lian dui an-quan biau-zhun de wen-ti mei-you fa-biau yi-jian

Soviet+ Union concerning safety standards of question have+not express opinion

'Soviet Union have not express opinion concerning question of safety standards'

3. Higher Syntactic Structures: Noun Phrases

One of the most important problems involving higher syntactic structures is that of the scope of noun-phrase modification. While most cases of ambiguous scope could be resolved semantically if enough knowledge of the world could be built into our system, we have found *syntactic* clues that enable us to handle such cases at a much earlier level of the analysis. The basic principle is that *a modifier always applies to the noun closest to it in the surface structure*. In terms of parsing, this is assured by moving the head noun node N_n up one node-level as soon as a modifier is conjoined to it. Since there are no rules allowing further modification at level N_{n+1}, unacceptable parses will abort at this point.

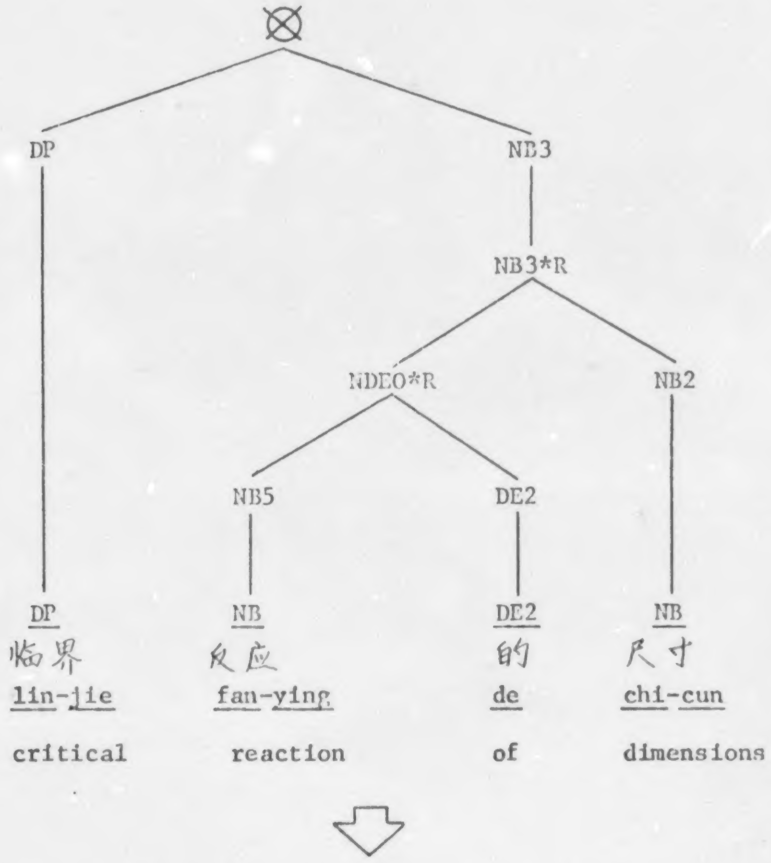
Following are some sets of structures illustrating problems of scope of modification. Within each set, (b) is the correct structure while (a) represents an unacceptable structure which aborts at a point (marked by Ⓝ) where there are no rules to incorrectly conjoin the relevant nodes. Note the cruciality of the analysis for the correct translation, as shown in the English outputs:

Figure 50:

Surface string:

临界	反应	的	尺寸
lin-jie	fan-ying	de	chi-cun
DP	NB	DE2	NB
critical	reaction	of	dimensions

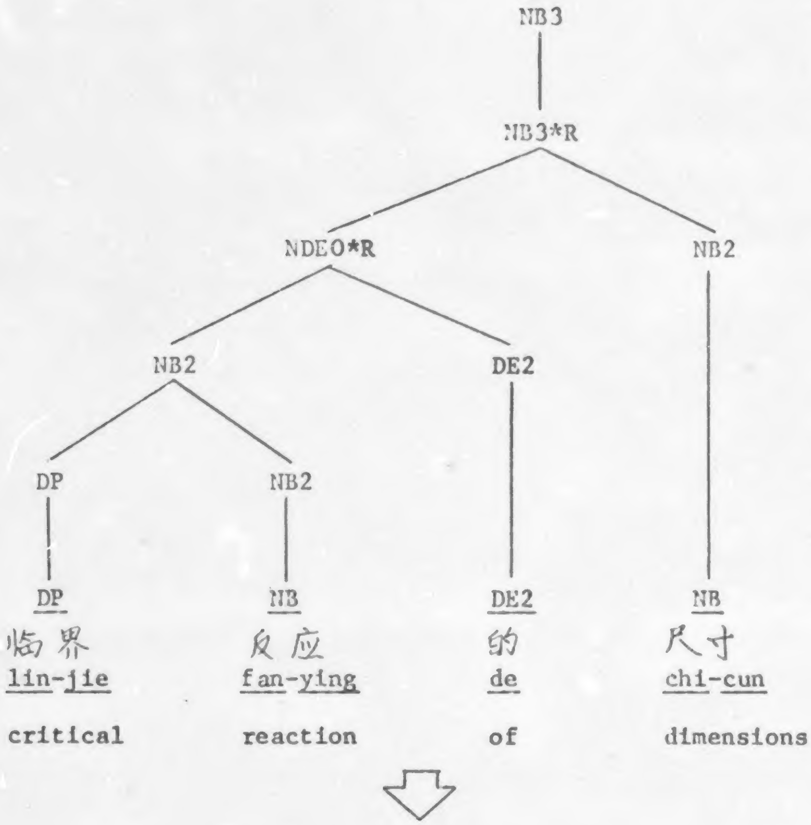
Structure (a): (unacceptable)


⊗
DP
NB3
NB3*R
NDEO*R
NB2
NB5
DE2
DP
NB
DE2
NB
临界
反应
的
尺寸
lin-jie
fan-ying
de
chi-cun
critical
reaction
of
dimensions

*'critical dimensions of reaction'

Figure 51:

Structure (b): correct


NB3
NB3*R
NDEO*R
NB2
NB2
DE2
DP
NB2
DP
NB
DE2
NB
临界
lin-jie
critical
反应
fan-ying
reaction
的
de
of
尺寸
chi-cun
dimensions

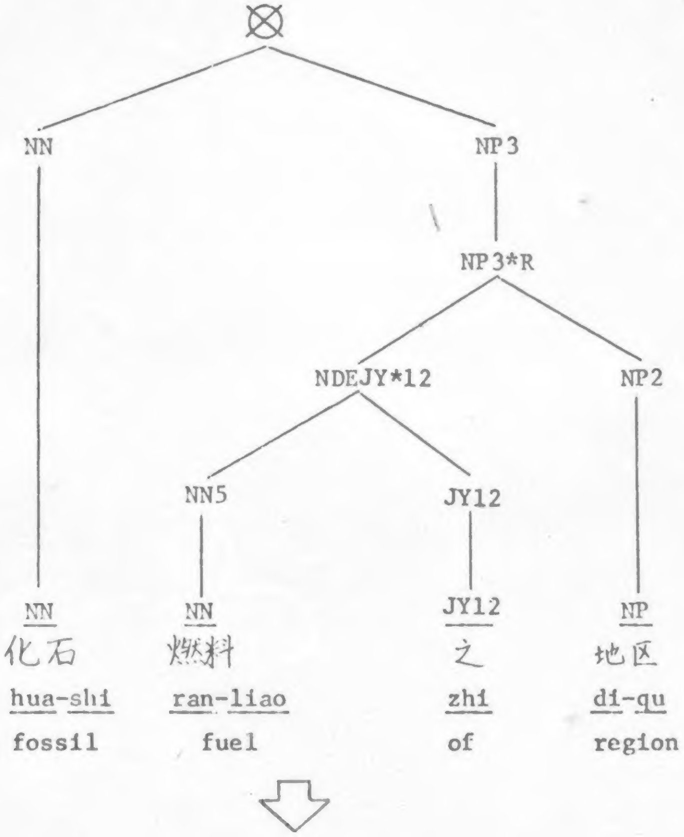
'dimensions of critical reaction'

Figure 52:

Surface string:

化石	燃料	之	地区
hua-shi	ran-liao	zhi	di-qu
NN	NN	JY12	NP
fossil	fuel	of	region

Structure (a): unacceptable


⊗
NN
NP3
NP3*R
NDEJY*12
NP2
NN5
JY12
NN
NN
JY12
NP
化石
燃料
之
地区
hua-shi
ran-liao
zhi
di-qu
fossil
fuel
of
region

*'fossil region of fuel'

Figure 53:

Structure (b): **correct**

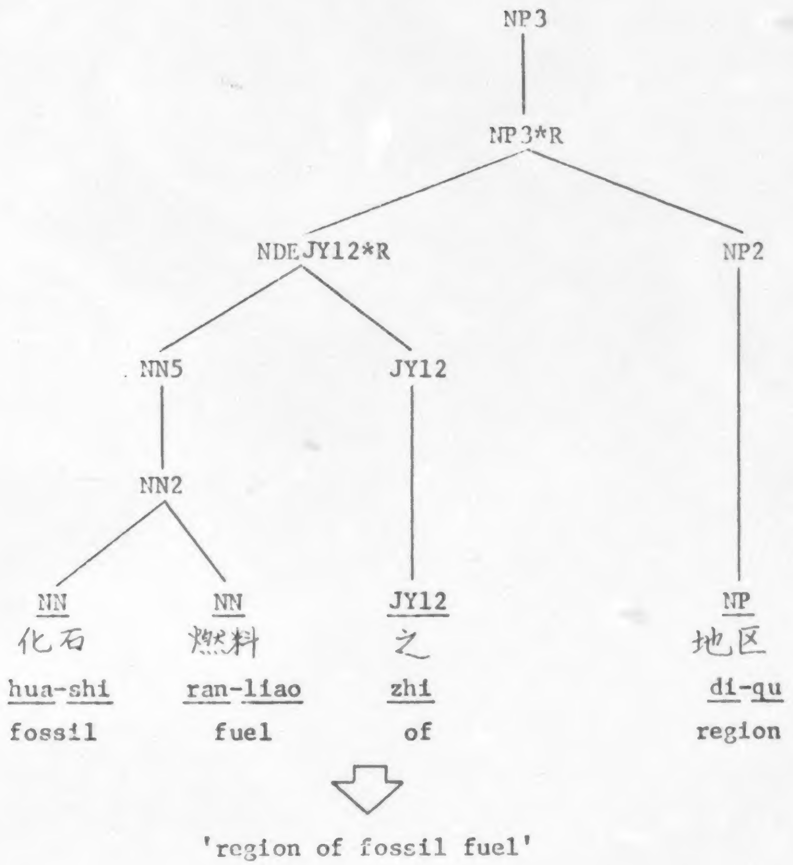
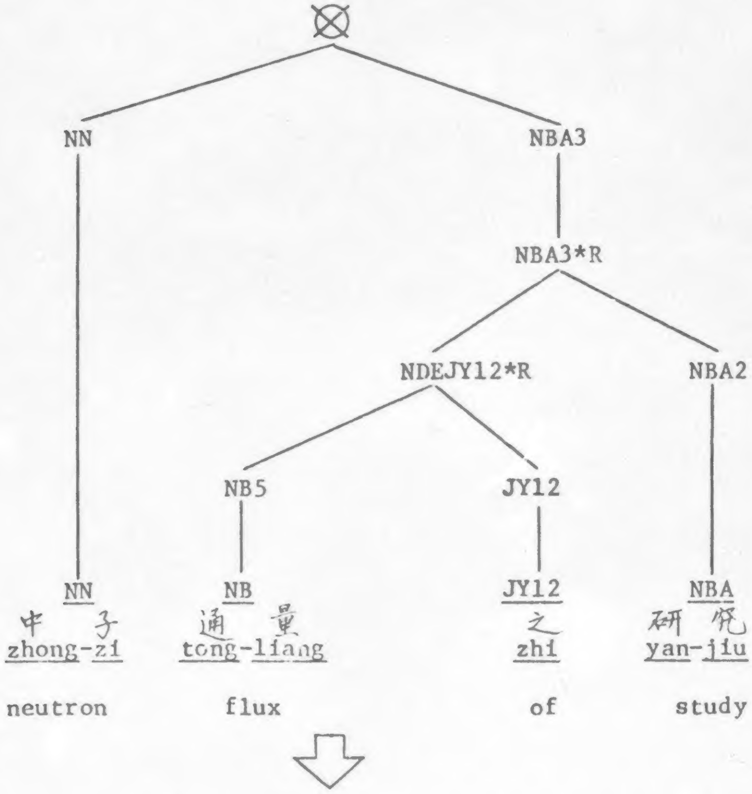

NP3
NP3*R
NDEJY12*R
NP2
NN5
JY12
NN2
NN
化石
hua-shi
fossil
NN
燃料
ran-liao
fuel
JY12
之
zhi
of
NP
地区
di-qu
region
'region of fossil fuel'

Figure 54:

Surface string:

中子	通量	之	研究
zhong-zi	tong-liang	zhi	yan-jiu
NN	NB	JY12	NBA
neutron	flux	of	study

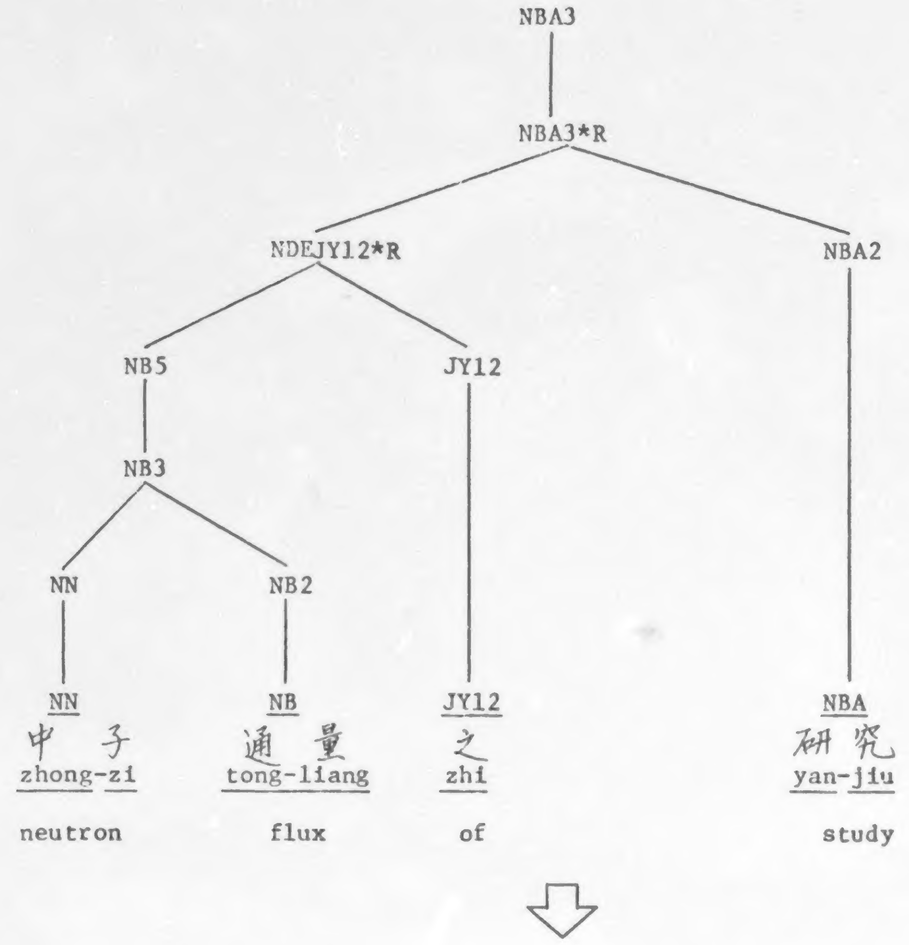
Structure (a): unacceptable


⊗
NN
NBA3
NBA3*R
NDEJY12*R
NBA2
NB5
JY12
NN
中子
zhong-zi
neutron
NB
通量
tong-liang
flux
JY12
之
zhi
of
NBA
研究
yan-jiu
study

*'neutron study of flux'

Figure 55:

Structure (b): correct


NBA3
NBA3*R
NDEJY12*R
NBA2
NB5
JY12
NB3
NN
NB2
NN
NB
JY12
NBA
中 子
zhong-zi
neutron
通 量
tong-liang
flux
之
zhi
of
研 究
yan-jiu
study

'study of neutron flux'

Figure 56:

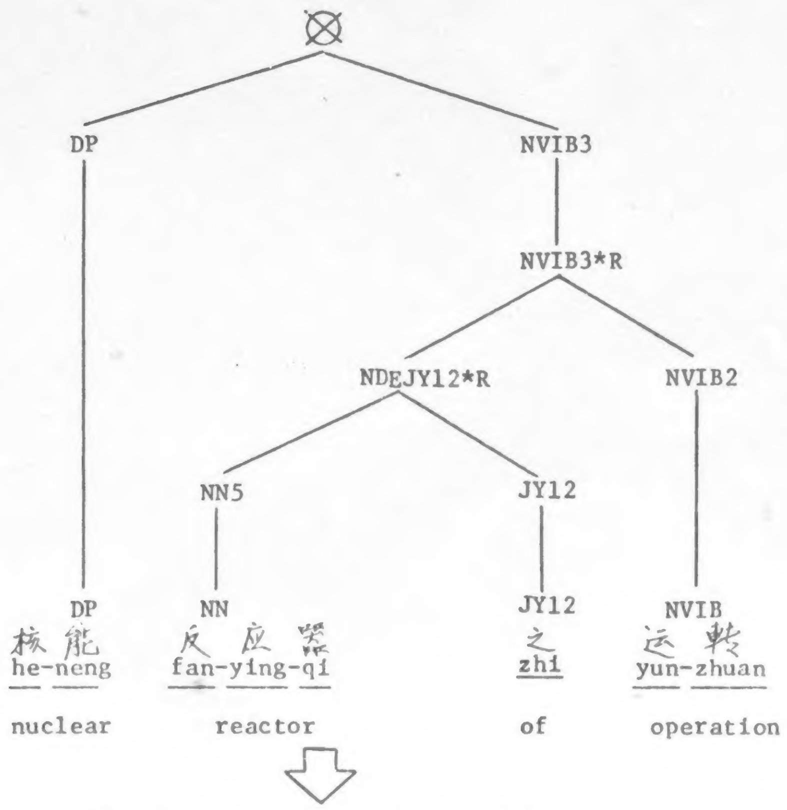
Surface string:

核能反应器之運轉
he-neng fan-ying-qi zhi yun-zhuan

DP NN JY12 NVIB

nuclear reactor of operation

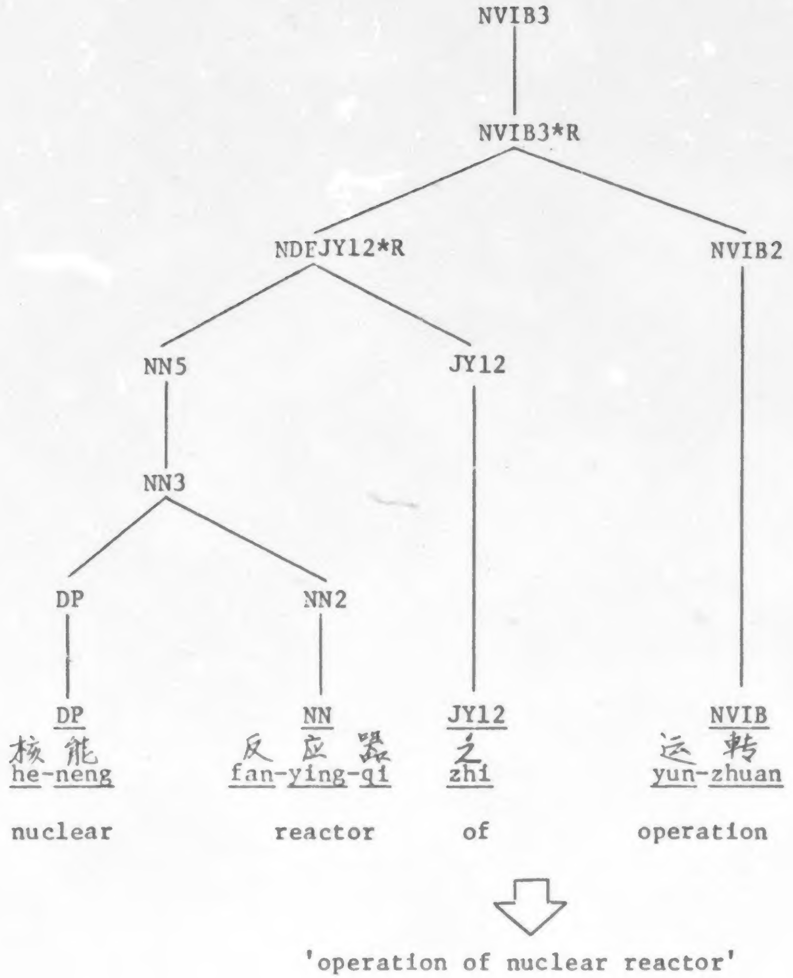
Structure (a): (unacceptable)


⊗
DP
NVIB3
NVIB3*R
NDEJY12*R
NVIB2
NN5
JY12
DP
NN
JY12
NVIB
核能
反应器
之
运轉
he-neng
fan-ying-qi
zhi
yun-zhuan
nuclear
reactor
of
operation

*'nuclear operation of reactor'

Figure 57:

Structure (b): correct


NVIB3
NVIB3*R
NDEJY12*R
NVIB2
NN5
JY12
NN3
DP
NN2
DP
NN
JY12
NVIB
核 能
he-neng
反 应 器
fan-ying-qi
之
zhi
运 转
yun-zhuan
nuclear
reactor
of
operation
'operation of nuclear reactor'

References

1. Chao, Y.R. 1968. A Grammar of Spoken Chinese. Berkeley and Los Angeles: University of California Press.

2. Cheung, Hung-nin Samuel. 1973. A comparative study in Chinese grammars: the *ba*-construction. *Journal of Chinese Linguistics* 1.3, 343-382.

3. Fan, Ji-yan. 1963. *Dong-ci he qu-xiang-xing hou zhi cheng-fen de jie-gou fen-shi* (Structural analysis of verbs and post-posed directional elements). *Zhongguo Yuwen* 2,136-160.

4. Frei,H. 1956. The ergative construction in Chinese: theory of Pekinese *ba*. *Gengo kenkyu* 31,22-50 and 32,83-115.

5. Hashimoto, A. Y. 1969. The verb 'to be' in modern Chinese. *Foundations of Language*, Suppl. ser. 9.4, 72-111.

6. Hashimoto, M. J. 1969. Observations on the passive construction. *Unicorn* 5,59-71.

7. Teng, Shou-hsin. 1972. A Semantic Study of Transitivity Relations in Chinese. Unpublished Ph.D. dissertation. University of California, Berkeley.

8. ________________ 1973. Transitivity and some problems with the *ba* construction in Mandarin Chinese. *Journal of Chinese Linguistics* 1.2, 208-221.

9. Wang, Huan. 1959. *Ba Zi Ju He Bei Zi Ju*. (Sentences with ba and bei). Jiaoyu Chubanshe, Peking.

10. Wang, W. S-Y., S.W. Chan et al. 1974. Development of Chinese-English Machine Translation System. Final Technical Report RADC-TR-74-22, (AD776813). Rome Air Development Center, Griffiss Air Force Base, New York.

V. CONCLUSION

The work undertaken during this contractual period has resulted in the implementation of a new series of programs for Chinese-English machine translation. Throughout this implementation great emphasis was placed on the requirement for machine-independence of the system. The state of the art in software design has pointed the way to this development. For a large, complex system such as ours, the cost of conversion to another machine would be prohibitive had the system been written with many machine dependencies. But the design of our system makes such conversion a comparatively routine or even trivial task. It is of course to be expected that the path of conversion will not be completely covered with roses, but every effort has been made so that the thorns can be easily spotted and removed.

At the close of this contract period, all modules for the QUINCE system were in existence and had been individually tested. Interfacing and testing these modules in a running text environment will be the work of our next effort. If manpower remains as in this contract, interfacing these modules is expected to be complete in a year's time, to be followed by actual tests on the completed system. In designing our system, we have anticipated the need for the system to be able to accommodate future expansions so as to keep pace with advances in linguistic research. It is not expected that components already implemented will have to be replaced in order to accomodate such advances. This is in conformity with our design criterion of flexibility. Since we have strived to program every module with clean interfaces, it would seem that although a system running in interactive or time-sharing mode was not part of our conception in design, the possibilities of converting to such a mode are much better than with our prototype SAS or other earlier systems making use of less 'structured' programming techniques. (But on no account should such a conversion be considered trivial.)

Since the new system has not yet been fully integrated, it is not now possible to make any realistic estimates of its throughput without being misleading. But as has been pointed out, the system is closely monitored during execution, so that optimization of the system can be carried out efficiently. At present, it seems that as far as running time is concerned, it is the various I/O processes rather than CPUL time which are economically significant on a comparative basis. As the system has been designed for a variety of machines, consequently the better the configuration of I/O and secondary storage hardware, the better it will execute. And since these routines have been written with timing included, the efficiency of different peripheral device configurations can be compared.

By making use of the dual SAS-QUINCE concept, we have been able to check out the major components of the system separately, and we can now rest assured that no major surprises will await us at the completion of system integration. We are especially happy with the fact that the basic linguistic tasks required for analysis of the Chinese sentence have been implemented as desired and that further refinements and additions to these linguistic tasks can be made incrementally in the system.

The English translations which the Berkeley system presently produces are by no means as smooth and as polished as desirable; there is obviously a great deal of room for improvement in this regard. Such English-specific problems as surface morphology lie beyond the scope of our present effort, which continues to consist primarily in obtaining for Chinese the best and least ambiguous analysis, and mapping the results of this analysis via interlingual transformations into English output in as correct a form as it is possible to achieve on the basis of the information retrievable from the Chinese analysis.

It is abundantly clear that there exist great structural differences between Chinese and English, and that a good deal of the information necessary to produce well-formed English surface structure cannot be derived from analysis of the Chinese; it is simply not there. Very often it is necessary to go beyond the confines of the Chinese sentence in search of some of this information, e.g., tense, number agreement, and scope of noun modification; and through careful and painstaking research we have been able to uncover some of this information for certain types of sentences. Yet great care must be taken in these cases, since over-hasty generalizations would serve only to nullify the results. For example, the distinction between definite and indefinite nouns is more often than not contextually determined (when it is not overtly indicated in the text), and therefore one must not assume that a definitized noun in one sentence will be equally definite in another sentence. Blindly assigning a definite article to the noun each time it occurs in text may well produce quite misleading output; and this task might better be performed by a human post-editor who can make such decisions on the basis of the context.

A great deal of useful data concerning problems particular to English is available in the research results of others (for instance, the University of Texas), and it is our hope that in the future it will be possible to incorporate some of this English-specific information into our system and thereby further improve the English output which it produces. It is unreasonable to expect to see in the very near future English output which requires no post-editing, yet one of our principal aims has been and will continue to be the minimizing of post-editing as much as possible. The system's output at the very least must be directly readable by the user, and must afford him a good grasp of the content of the original Chinese input; this is a minimal requirement which we expect our machine translation system to meet.

Unless it incorporates a vast information-retrieval component and a sentence-understanding component, a machine translation system should not be expected to resolve contextual problems very much beyond the bounds of a paragraph -- and a pretty short paragraph at that. Although our system can deal successfully with consecutive sentences in a text one by one and produce an English output string for each sentence, it is still too soon to expect it to be capable of linking together these separate sentence-by-sentence analyses to deal with them as a whole. One well-formed paragraph might assume that certain statements are understood and certain conclusions drawn; but then the next paragraph might be totally unrelated, requiring a great deal of deductive power to bridge the gap from one paragraph to the next. Some research in artificial intelligence is currently focused on this problem, but in order to facilitate (or even make possible) the designing of experimental systems, the universes of discourse in which such deductions are being made have been intentionally impoverished and severely restricted. Such stringent controls and limitations cannot be imposed in the context of machine translation, however, since it is necessary to deal with the infinitely large universe of real language.

During the past two years, the Berkeley Project has completed design and coding of the modular components of a basic new system, the QUINCE system, which has a great deal of flexibility to allow for future expansion. Over the next few years we expect to continue to augment and improve the capabilities of the QUINCE system along with advances in linguistic research to treat ever-increasing numbers of syntactic and semantic problems of complex Chinese sentences and produce even better English output.

Appendix 1. Computer-Plotted Displays of Levels of Chinese Sentence Structure

Figures A1 through A5 represent the successive application of different grammars to the example sentence and the resulting intermediate structures which are available for visual inspection by the linguist and lexicographer.

The input string for the example sentence in telecodes is:

原	子	核	在	原	子	中	占	的	地	位
0626	1311	2702	0961	0626	1311	0022	0594	4104	0966	0143
很	小									
1771	1420									

Each telecode is numbered from left to right with a 'sentence position' for identification. The accompanying table is the system's sentence dictionary used in processing the above sentence; each of its lexical entries contains a code indicating the grammatical function of the entry and an English gloss. This same information is also to be seen at the bottom of Figure A1, which shows the first or lowest level of analysis by the system. The dashed branches represent the same Chinese character (telecode) string with the indicated categories. Thus zai 在 (0961) has been assigned five separate grammar codes reflecting a variety of different functions demanded by different English surface strings. Similarly, de 的 (4104) has eight possible realizations in English depending on the particular structure in which it

Table 1. Sentence Dictionary

telecode string	grammar code	English gloss
062613112702	NN	nucleus
0961	VG14	xxxx[no gloss]
0961	VG13	at/in
0961	VG12	be
0961	VG11	be+at/be+in
0961	APR	be+[verb]+ing
06261311	NN	atom
0022	NL16	the+middle+of/the+interior+of
0022	NL15	xxxx[no gloss]
0022	NL12	in/among/in+the+middle+of
0022	NF11	China
0022	DP	intermediate/medium
0594	VT	occupy
4104	2X-AZT	xxxx[no gloss]
4104	DE7	that+of/those+of
4104	DE6	that+which/those+which/what
4104	DE5	who
4104	DE4	xxxx[no gloss]
4104	DE3	which/that
4104	DE2	of
4104	DE1	's
09660143	NBY	position
1771	AQ	very
1420	VQ12	small

occurs in Chinese. The slashes between alternative glosses are replaced by * in the plots. Note that if these alternative glosses are also counted, then of course we have more than 8 realizations. But these alternatives cannot be adequately distinguished at present merely from a surface analysis of the Chinese structure. They are much more relevant to the English surface structure. In a purely Chinese context of analysis some of these differentiations for zai and de would result in over differentiation.

Looking at Figure A1 again, it should be pointed out that the many branching lines converging to a top node are an artifact of the current system. It does not mean any complete tree structure has been formed at that stage. If the plotted display along such a node is dominated by a number only, then no rule has been applied to form a real node. It is only when a node is dominated by a category symbol plus a number underneath that a real node has been formed. Thus in Figure A1, no rule in the grammar has actually applied. (All the dominating nodes indicated are just visual repetitions of the terminal symbols - joined by a vertical line.)

In Figure A2, only the rightmost two constituents were formed into a higher node by the rule

$$VIQ2 \longrightarrow AQ + VQ12$$

No other rule in grammar 2 has applied.

In Figure 3A there is a sudden "explosion of activity" involving rules in grammar 3. Practically all the constituents became identified except for the topmost two constituents. This means no rule for putting

NB2*R and VIQ2 together was available in grammar 3.

Nor is this rule available in grammar 4 (see Figure A4).

In Figure A5, the desired rule

$$\text{IND} \longrightarrow \text{NB2*R} + \text{VIQ2}$$

was available, completing the analysis of the sentence.

Figure A3b is derived from Figure A3a by interlingual restructuring of the analysis represented by Figure A3a. Figure A3a is the available analysis at that stage for the Chinese structure. Figure A3b restructures this to arrive at the surface structure order of its English equivalent - less any extra morphological processes required in English. So even at the Figure A3 stage, we have obtained the subject ("position which/that nucleus occupy in/among/in the middle of atom") and the predicate ("very small").

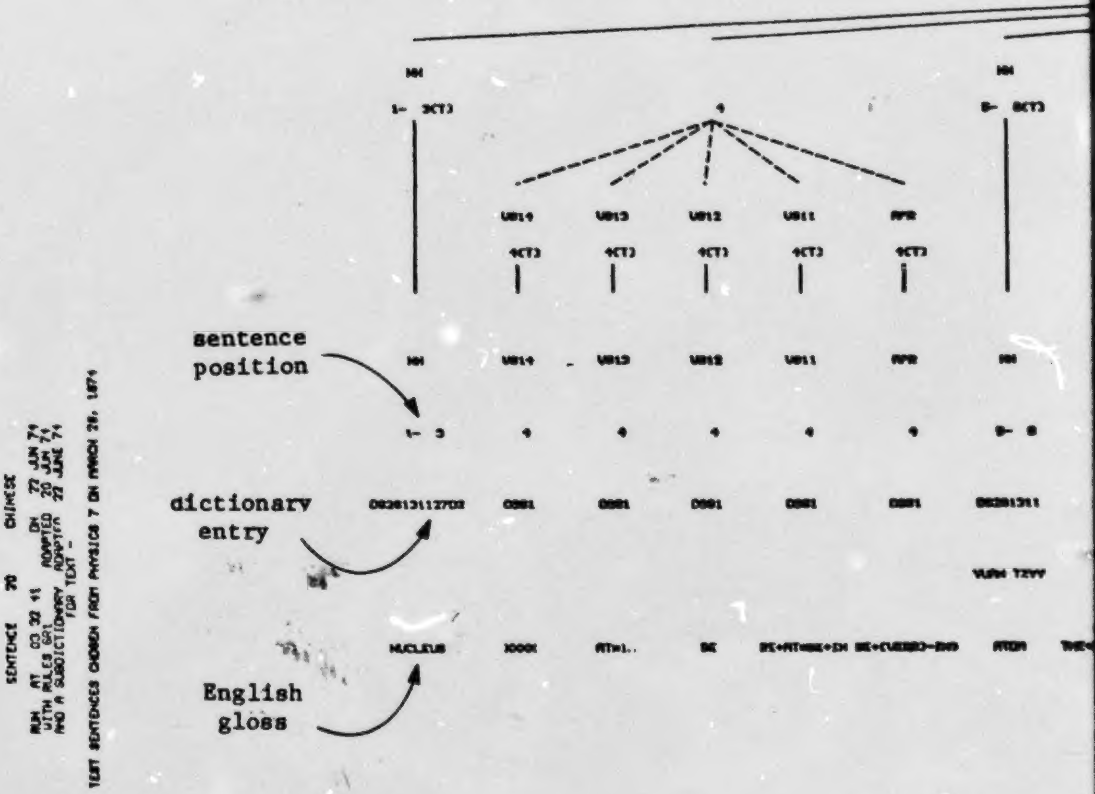
The last stage (Figure A5) tells us that an actual sentence has been recognized by the grammar, as indicated by a single top node labeled IND.

The English string output that is available after this stage 5 is:

> position which/that nucleus occupy in/among/in
> the middle of atom very small

To obtain smooth, stylistic English, post-editing is envisioned at this point. For example, the articles, tense and number agreement will have to be supplied. The copula in this particular structure can be inserted by means of a specific interlingual rule which inspects the predicate for verbs of the stative type. As the stock of such rules is increased with additional research, the post-editing function will be taken over more and more by the computer.

Finally, it should be pointed out that the analysis for this sentence produced by the system was unambiguous. (Ambiguous substructures would have shown up on these plots as dashed branches dominated by some node--analogous to those described for Figure A1.) The new QUINCE system being implemented will enable us to systematically enhance the capabilities for obtaining better analyses and recognizing the correct structures for a Chinese sentence, which of course will lead to overall improvements in translation.


sentence position
dictionary entry
English gloss
SENTENCE 70 CHINESE
RUN AT 03 32 41 ON 22 JUN 74
WITH RULES 601 ADAPTED 20 JUN 74
AND A SUBDICTIONARY ADAPTED 22 JUNE 74
FOR TEXT -
TEST SENTENCES CHOSEN FROM PHYSICS 7 ON MARCH 28, 1974
NUCLEUS
ATOM

INPUT STRING 0626 131

SENTENCE POSITION 001 00

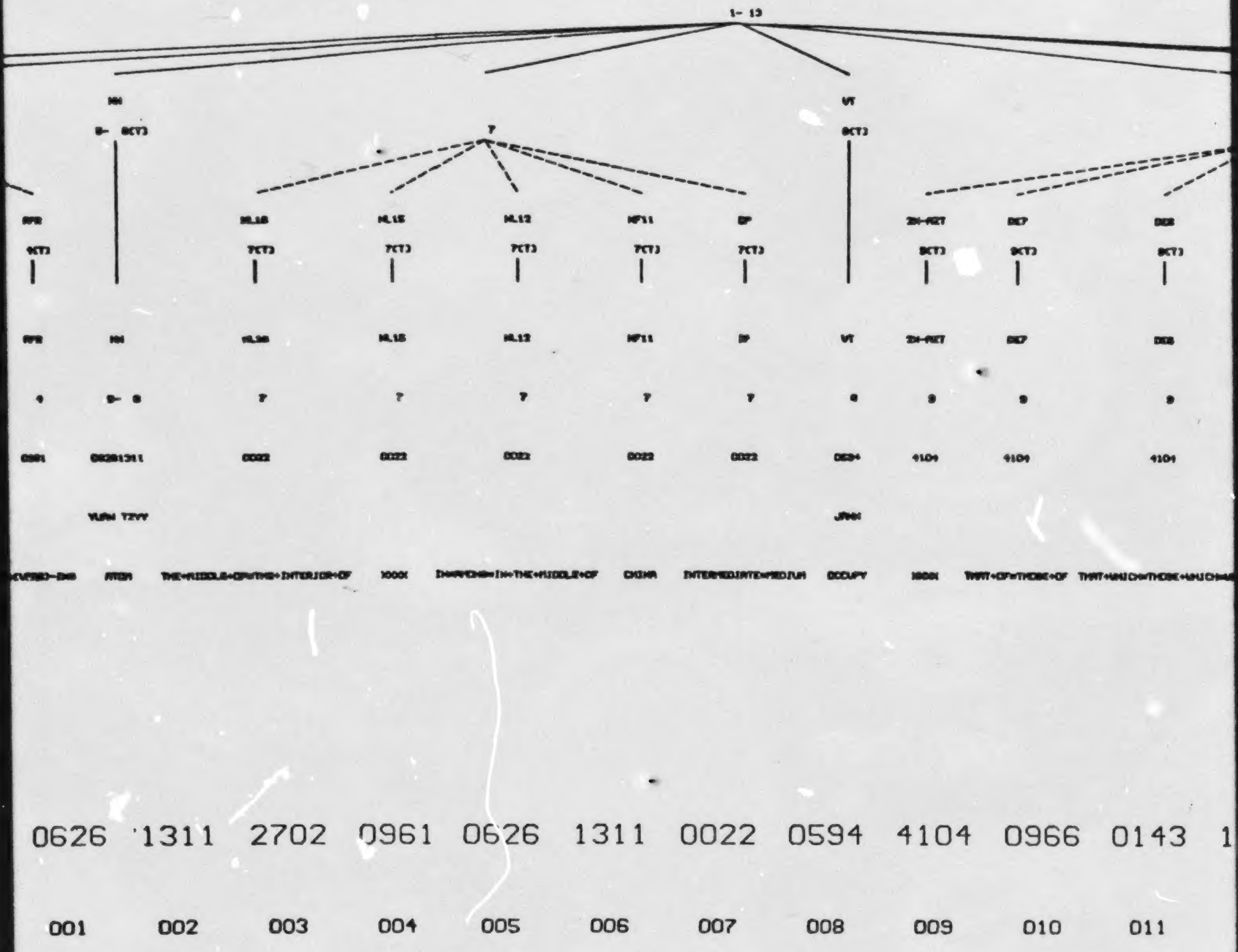
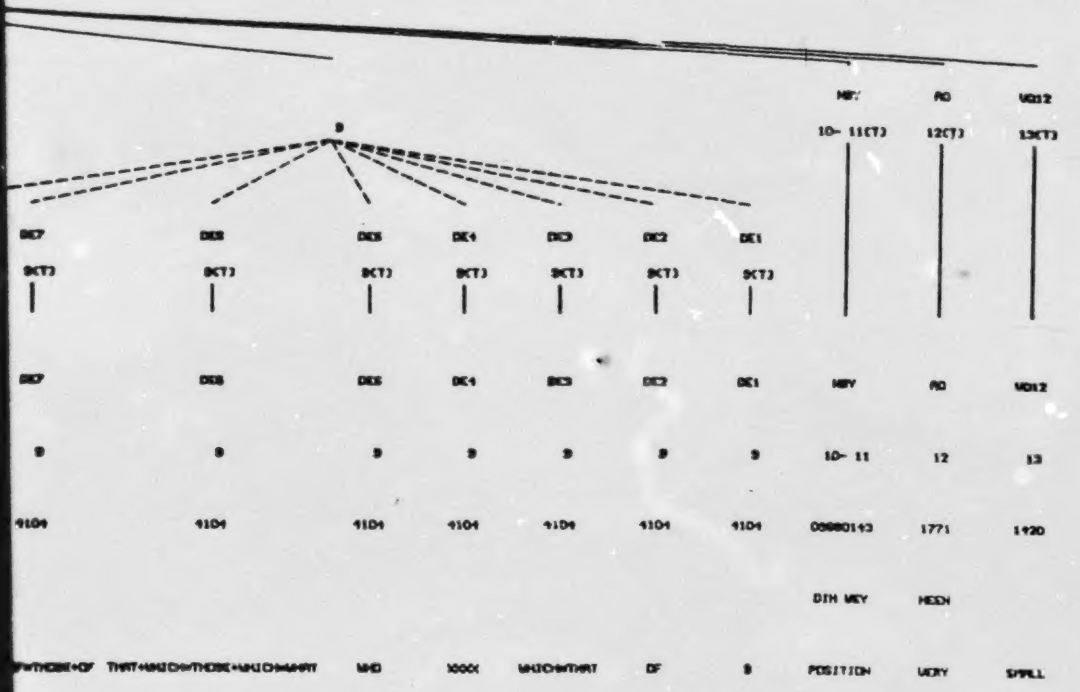

1- 13
VT
OCCUPY
CHINA
INTERMEDIATE+MEDIUM
THAT+OF=THOSE+OF

Figure A1


9
DE7
DE6
DE5
DE4
DE3
DE2
DE1
9(T)
NBY
AD
VD12
10- 11(T)
12(T)
13(T)
DE7
DE6
DE5
DE4
DE3
DE2
DE1
NBY
AD
VD12
9
10- 11
12
13
4104
00000143
1771
1420
DIH WEY
HEEN
+THOSE+OF
THAT+WHICH+THOSE+WHICH+WHAT
WHO
XXXX
WHICH+THAT
OF
9
POSITION
VERY
SMALL

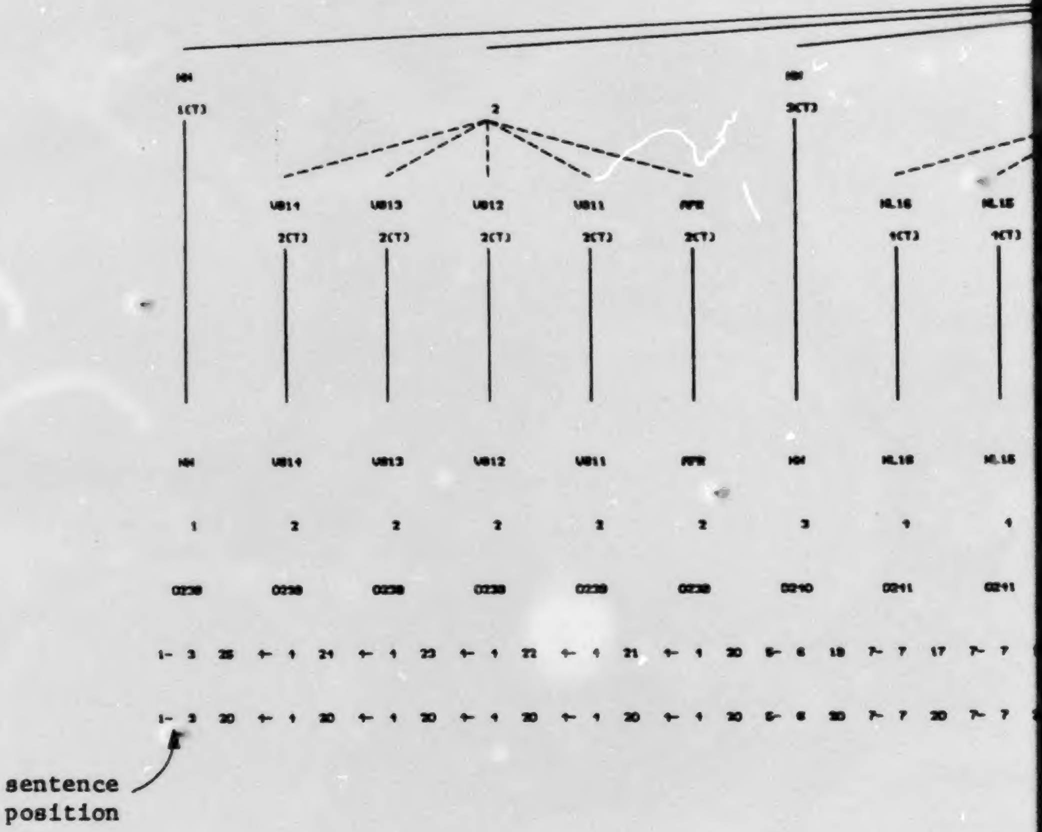
SENTENCE 20 CHINESE
RUN AT 03 32 41 ON 22 JUN 74
WITH RULES OR1 ADAPTED 20 JUN 74
AND A SUBDICTIONARY ADAPTED 22 JUNE 74
FOR TEXT -
TEST SENTENCES CHOSEN FROM PHYSICS 7 ON MARCH 20, 1974

0966 0143 1771 1420

010 011 012 013

206 A

SENTENCE 20 CHINESE
RUN AT 06 03 62 ON 22 JUN 74
WITH RULES 6R2 ADAPTED 20 JUN 74
AND A SUBDICTIONARY ADAPTED 22 JUNE 74
FOR TEXT -
TEST SENTENCES CHOSEN FROM PHYSICS 7 ON MARCH 28, 1974


NN
1(T)
2
VB14
2(T)
VB13
2(T)
VB12
2(T)
VB11
2(T)
APR
2(T)
NN
3(T)
NL16
4(T)
NL15
4(T)
NN VB14 VB13 VB12 VB11 APR NN NL16 NL15
1 2 2 2 2 2 3 4 4
0238 0238 0238 0238 0238 0238 0240 0241 0241
1- 3 25 4- 4 24 4- 4 23 4- 4 22 4- 4 21 4- 4 20 5- 6 19 7- 7 17 7- 7
1- 3 20 4- 4 20 4- 4 20 4- 4 20 4- 4 20 4- 4 20 5- 6 20 7- 7 20 7- 7
sentence
position

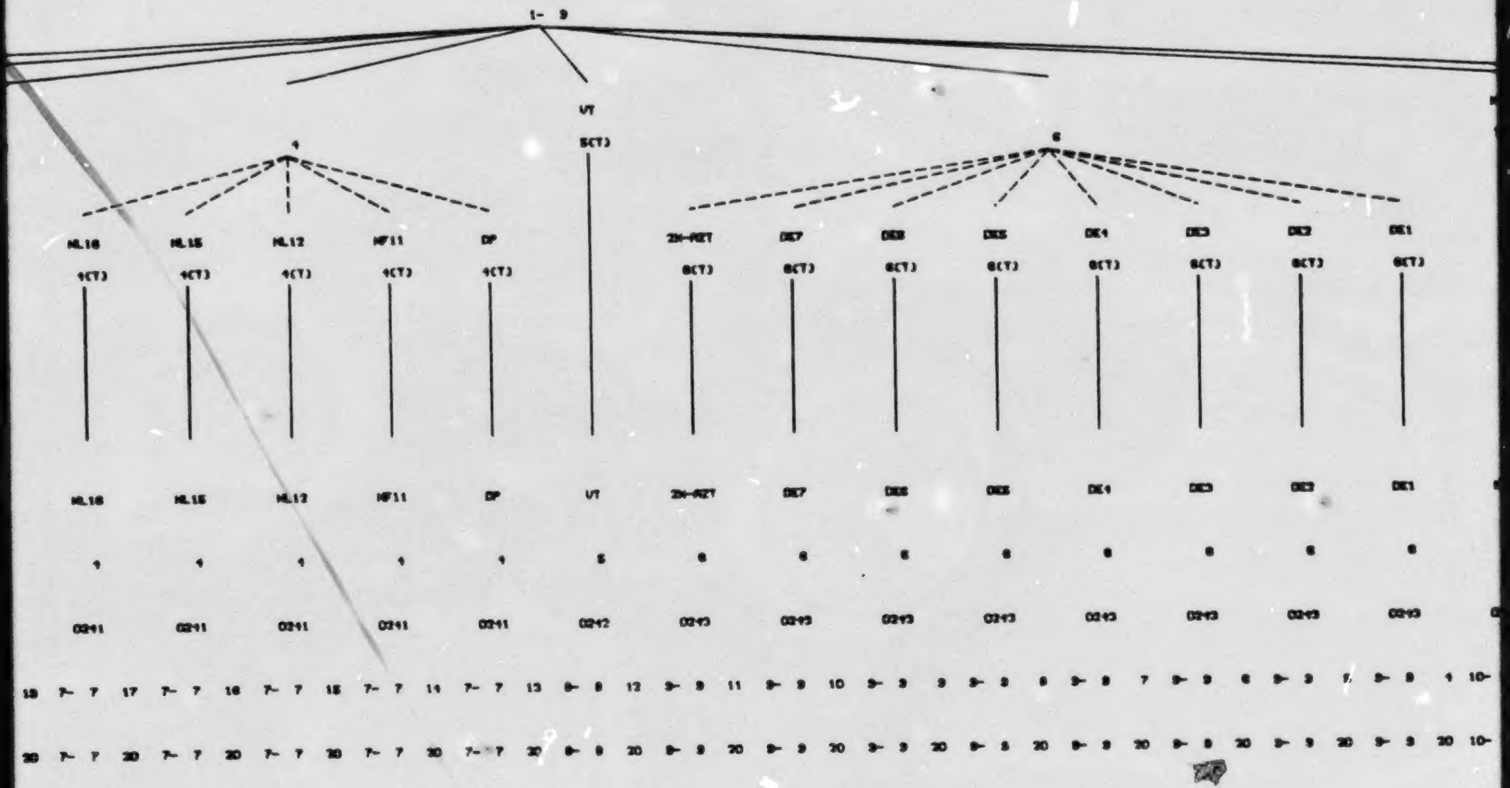
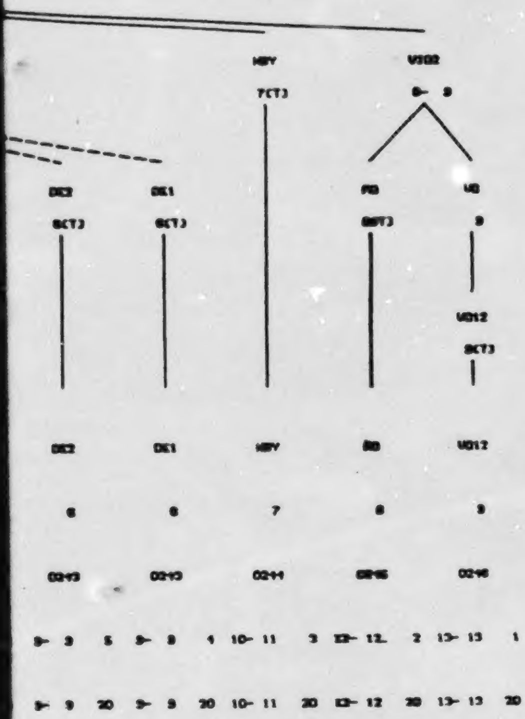

1- 9
VT
5(T)
4
6
HL16
HL15
HL12
HF11
DP
ZN-RZT
DE7
DE6
DE5
DE4
DE3
DE2
DE1
4(T)
6(T)
0241
0242
0243
18 7- 7 17 7- 7 16 7- 7 15 7- 7 14 7- 7 13 9- 9 12 9- 9 11 9- 9 10 9- 9 9 9- 9 8 9- 9 7 9- 9 6 9- 9 5 9- 9 4 10-
20 7- 7 20 7- 7 20 7- 7 20 7- 7 20 7- 7 20 9- 9 20 9- 9 20 9- 9 20 9- 9 20 9- 9 20 9- 9 20 9- 9 20 9- 9 20 9- 9 20 10-

Figure A2

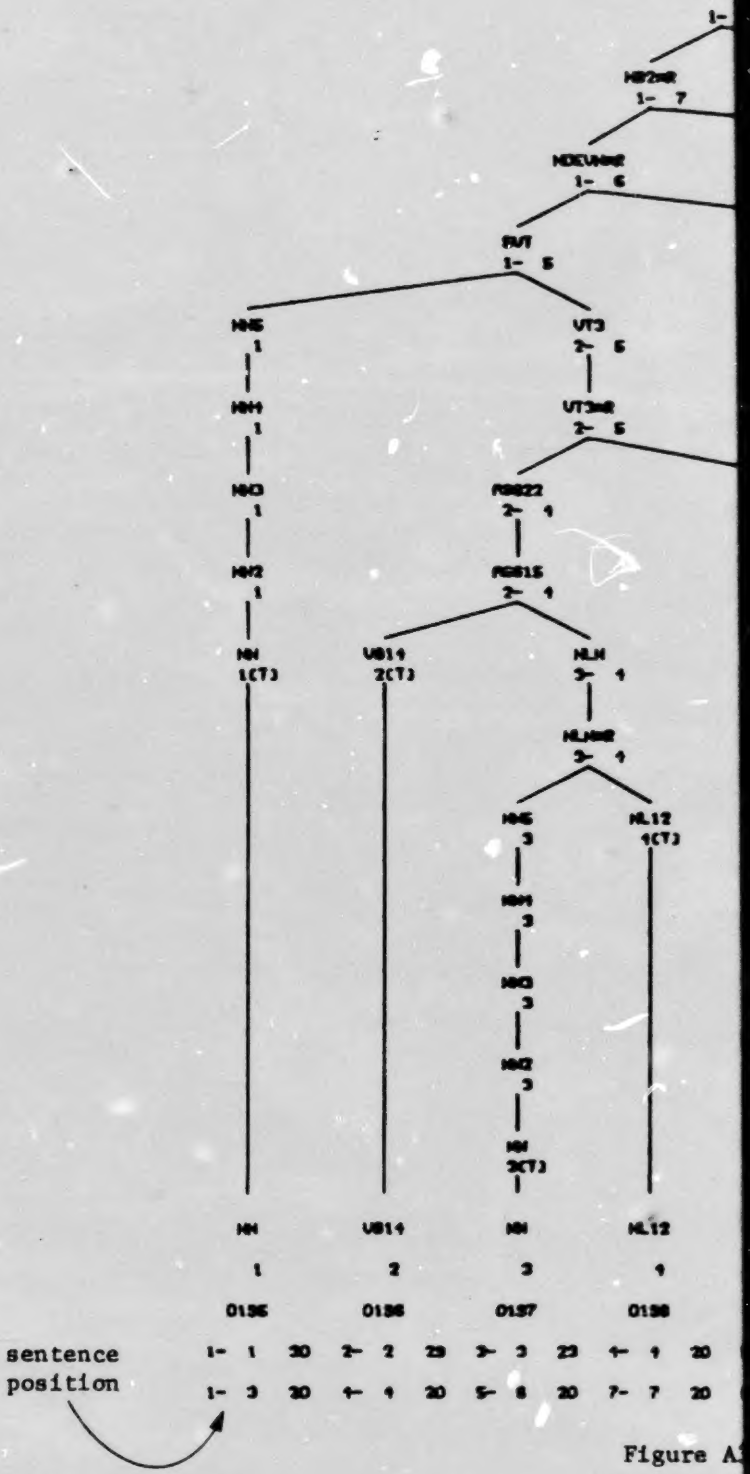

WBY
7(T)
VSO2
8- 9
DE2
6(T)
DE1
6(T)
RD
8(T)
VD
9
VD12
9(T)
DE2
DE1
WBY
RD
VD12
6
6
7
8
9
0243
0243
0244
0245
0246
9- 9 5 9- 9 4 10- 11 3 12- 12 2 13- 13 1
9- 9 20 9- 9 20 10- 11 20 12- 12 20 13- 13 20

SENTENCE. 20 CHINESE
RUN AT 06 03 52 ON 22 JUN 74
WITH RULES BR2 ADAPTED 20 JUN 74
AND A SUBDICTIONARY ADAPTED 22 JUNE 74
FOR TEXT -
TEST SENTENCES CHOSEN FROM PHYSICS 7 ON MARCH 28, 1974

SENTENCE 20 CHINESE

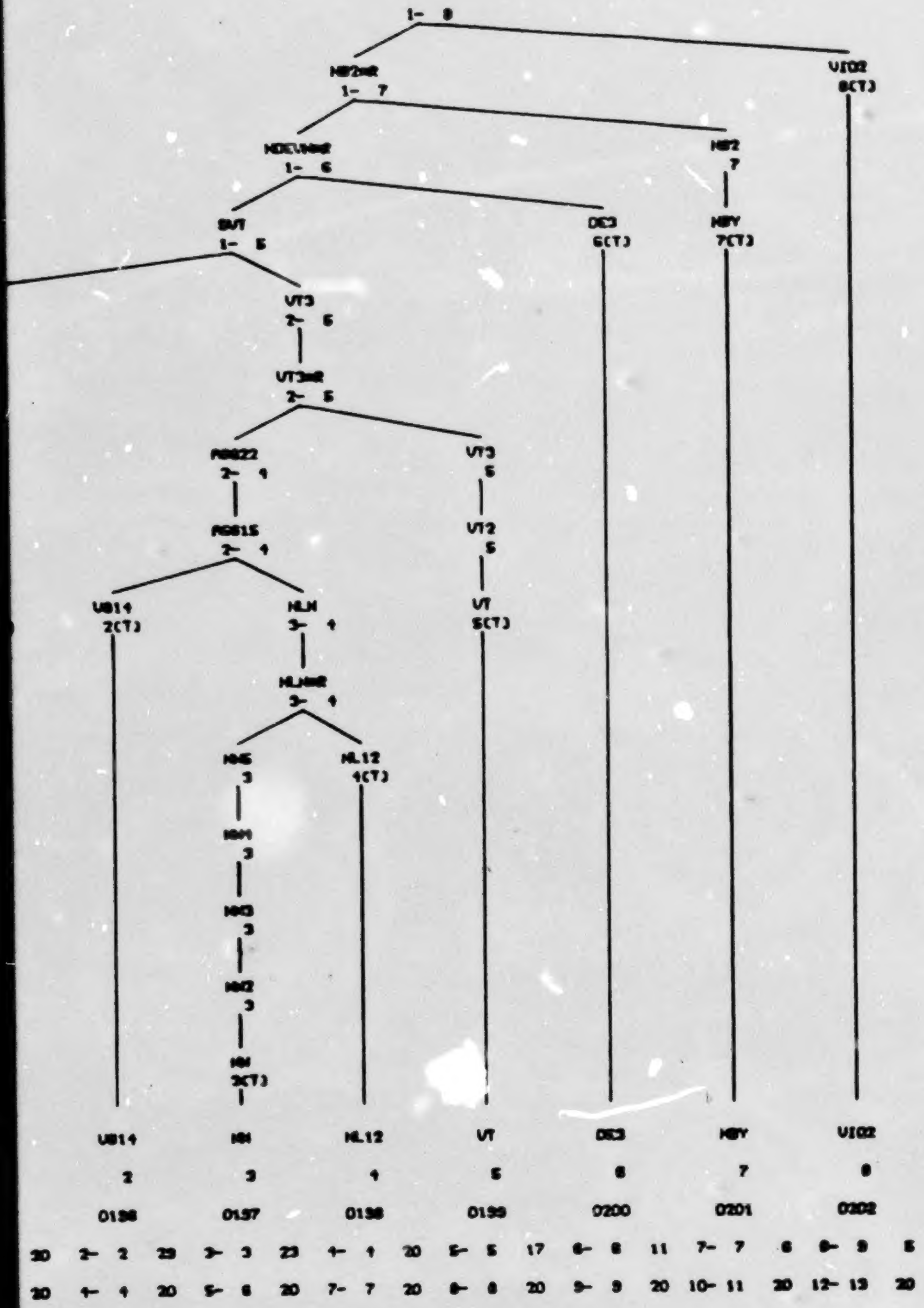
RUN AT 00 57 34 ON 23 JUN 74
WITH RULES GRG ADAPTED 20 JUN 74
AND A SUBDICTIONARY ADAPTED 22 JUNE 74
FOR TEXT -

TEST SENTENCES CHOSEN FROM PHYSICS 7 ON MARCH 29, 1974



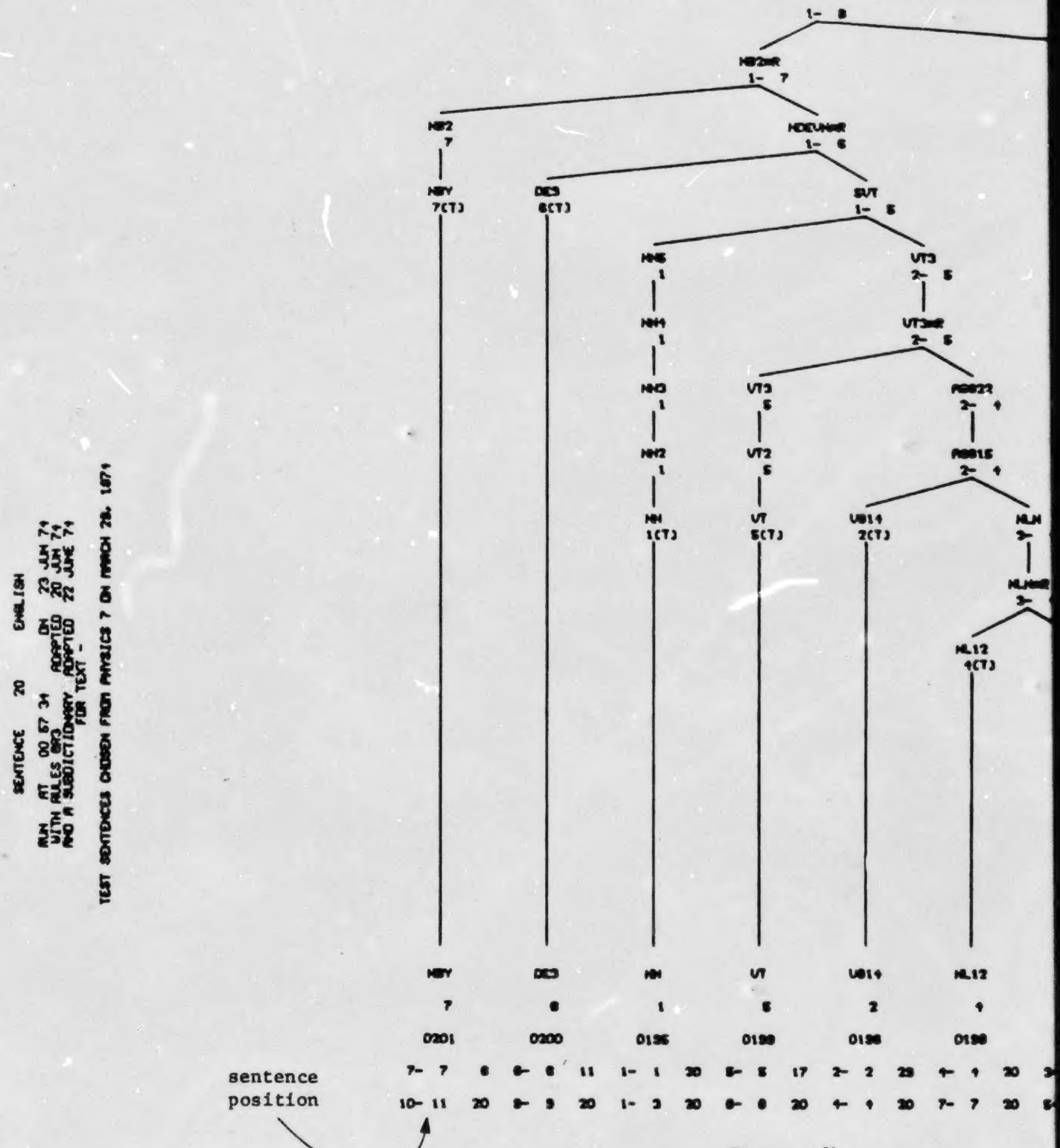
1-
HB2mR
1- 7
HDEUmmR
1- 6
RVT
1- 5
NN5
1
VT3
2- 5
NN4
1
VT3mR
2- 5
NN3
1
RB822
2- 4
NN2
1
RB815
2- 4
NN
1(T)
VB14
2(T)
NLN
3- 4
NLmmR
3- 4
NN5
3
NL12
4(T)
NN4
3
NN3
3
NN2
3
NN
3(T)
NN
VB14
NN
NL12
1
2
3
4
0136
0136
0137
0138
sentence position
1- 1 20 2- 2 23 3- 3 23 4- 4 20
1- 3 20 4- 4 20 5- 6 20 7- 7 20

Figure A


1- 8
NP2MR
1- 7
VI02
8(T)
NDEVMMR
1- 6
NP2
7
SVT
1- 5
DE3
6(T)
NBY
7(T)
VT3
2- 5
VT3MR
2- 5
ADB22
2- 4
VT3
5
ADB15
2- 4
VT2
5
UB14
2(T)
NLN
3- 4
VT
5(T)
NLNMR
3- 4
NN5
3
NL12
4(T)
NN4
3
NN3
3
NN2
3
NN
3(T)
UB14 NN NL12 VT DE3 NBY VI02
2 3 4 5 6 7 8
0136 0137 0138 0139 0200 0201 0202
20 2- 2 23 3- 3 23 4- 4 20 5- 5 17 6- 6 11 7- 7 6 8- 9 5
20 1- 4 20 5- 6 20 7- 7 20 8- 8 20 9- 9 20 10- 11 20 12- 13 20

SENTENCE 20 CHINESE
RUN AT 00 57 34 ON 23 JUN 74
WITH RULES BP3 ADAPTED 20 JUN 74
AND A SUBDICTIONARY ADAPTED 22 JUNE 74
FOR TEXT -
TEST SENTENCES CHOSEN FROM PHYSICS 7 ON MARCH 28, 1974

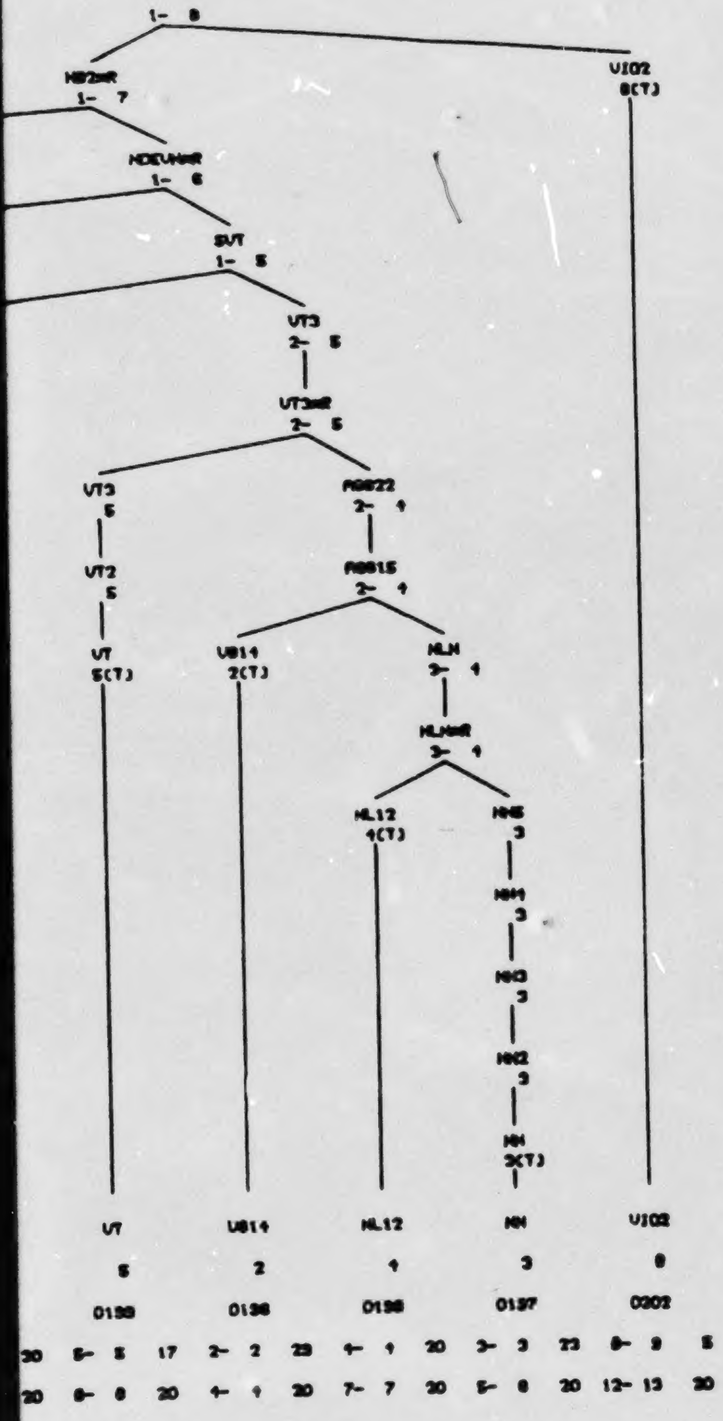
Figure A3a



```
SENTENCE    20       ENGLISH
RUN   AT   00 57 34            ON    23 JUN 74
WITH RULES 093        ADAPTED  20 JUN 74
AND A SUBDICTIONARY   ADAPTED  22 JUNE 74
                FOR TEXT -
TEST SENTENCES CHOSEN FROM PHYSICS 7 ON MARCH 26, 1974
```

1- 8
NB2MR
1- 7
NB2
7
NDEVNMR
1- 6
NBY
7(T)
DE3
8(T)
SVT
1- 5
NN5
1
VT3
2- 5
NN4
1
VT3MR
2- 5
NN3
1
VT3
5
AB022
2- 4
NN2
1
VT2
5
AB015
2- 4
NN
1(T)
VT
5(T)
VB14
2(T)
NLN
3-
NLNMR
3-
NL12
4(T)
NBY
DE3
NN
VT
VB14
NL12
7
8
1
5
2
4
0201
0200
0196
0199
0198
0199
7- 7 8 8- 8 11 1- 1 20 5- 5 17 2- 2 23 4- 4 20
10- 11 20 8- 9 20 1- 3 20 6- 6 20 4- 4 20 7- 7 20
sentence position

Figure A3b

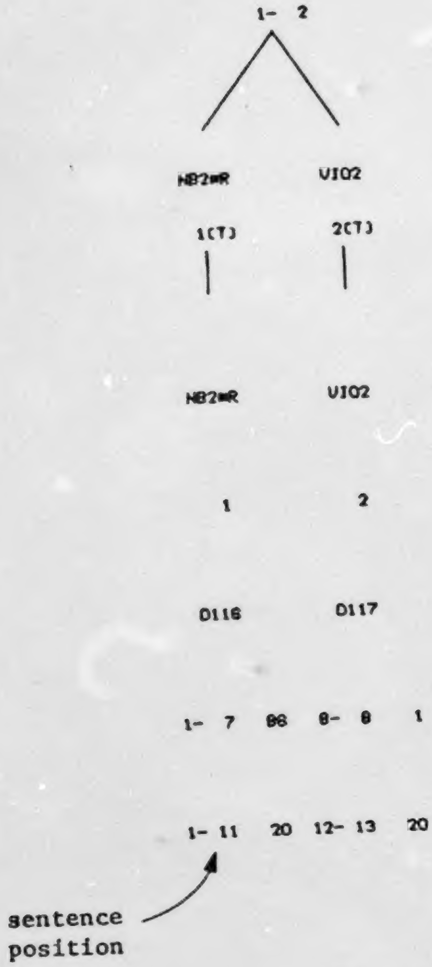

1- 8
NB2HR
1- 7
VI02
8(T)
NDEVHHR
1- 6
SVT
1- 5
VT3
2- 5
VT3HR
2- 5
VT3
5
AB022
2- 4
VT2
5
AB015
2- 4
VT
5(T)
VB14
2(T)
NLN
3- 4
NLNHR
3- 4
NL12
4(T)
NN5
3
NN4
3
NN3
3
NN2
3
NN
3(T)
VT
5
0139
VB14
2
0138
NL12
4
0138
NN
3
0137
VI02
8
0202
30 5- 5 17 2- 2 23 4- 4 20 3- 3 23 8- 9 5
20 8- 8 20 4- 4 20 7- 7 20 5- 6 20 12- 13 20

SENTENCE 20 ENGLISH
RUN AT 00 57 34 ON 23 JUN 74
WITH RULES 6N3 ADOPTED 20 JUN 74
AND A SUBDICTIONARY ADOPTED 22 JUNE 74
FOR TEXT -

TEST SENTENCES CHOSEN FROM PHYSICS 7 ON MARCH 28, 1974

Figure A3b

```
SENTENCE      20        CHINESE
RUN      AT   01 49 47        ON    23 JUN 74
  WITH RULES GR4     ADAPTED   20 JUN 74
  AND A SUBDICTIONARY  ADAPTED  22 JUNE 74
           FOR TEXT -
TEST SENTENCES CHOSEN FROM PHYSICS 7 ON MARCH 28. 1974
```

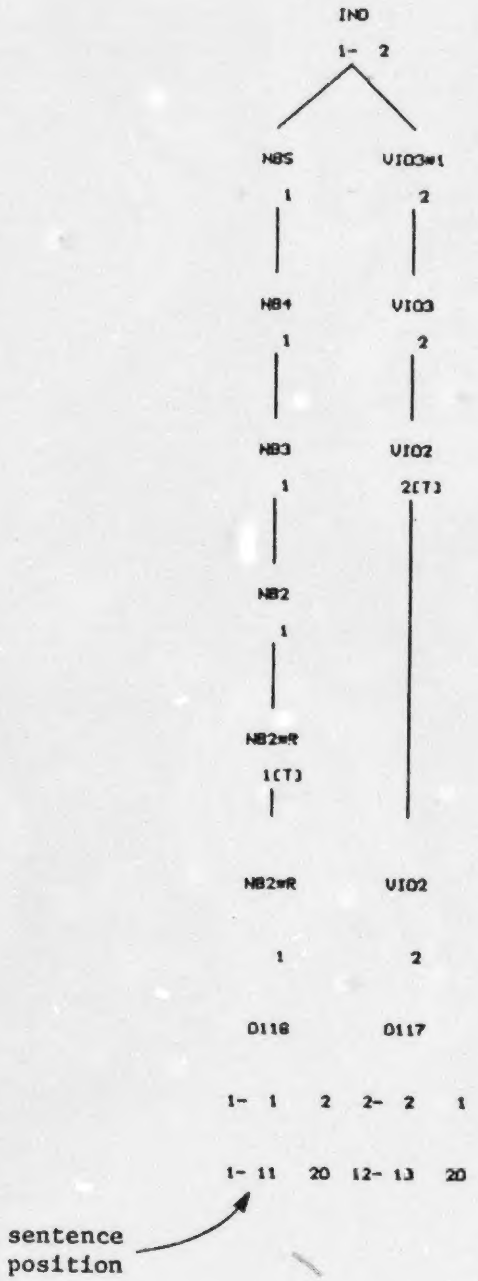

1- 2
NB2*R
VIO2
1(T)
2(T)
NB2*R
VIO2
1
2
D116
D117
1- 7
86
8- 8
1
1- 11
20
12- 13
20
sentence position

```
SENTENCE      20        CHINESE
RUN      AT   01 49 47        ON    23 JUN 74
  WITH RULES GR4     ADAPTED   20 JUN 74
  AND A SUBDICTIONARY  ADAPTED  22 JUNE 74
           FOR TEXT -
TEST SENTENCES CHOSEN FROM PHYSICS 7 ON MARCH 28. 1974
```

Figure A4

SENTENCE 20 CHINESE IND FRAME 21
RUN AT 02 52 49 ON 23 JUN 74
WITH RULES GRS ADAPTED 20 JUN 74
AND A SUBDICTIONARY ADAPTED 22 JUNE 74
FOR TEXT -

TEST SENTENCES CHOSEN FROM PHYSICS 7 ON MARCH 28, 1974


IND
1- 2
NB5
VIO3*1
1
2
NB4
VIO3
1
2
NB3
VIO2
1
2(T)
NB2
1
NB2*R
1(T)
NB2*R
VIO2
1
2
D116
D117
1- 1 2 2- 2 1
1- 11 20 12- 13 20
sentence position

SENTENCE 20 CHINESE IND FRAME 21
RUN AT 02 52 49 ON 23 JUN 74
WITH RULES GRS ADAPTED 20 JUN 74
AND A SUBDICTIONARY ADAPTED 22 JUNE 74
FOR TEXT -

TEST SENTENCES CHOSEN FROM PHYSICS 7 ON MARCH 28, 1974

Figure A5

Appendix 2: Style Differences in Modern Written Scientific Text

The excerpt of Chinese text in this appendix is an almost extreme example of the intermixing of modern and classical writing styles within a single short span of text. The four paragraphs of text are from a monograph on atomic energy. It is not necessary to go into a detailed analysis to get a feeling of the style mixture; a cursory reading is sufficient. In fact, a rough quantitative indication of this difference can be given with regard to these four paragraphs. The first paragraph is the end of a previous section and the next three paragraphs begin a new section. Paragraph 1 has a total character count of 334 (鈾-235 'uranium-235' is counted as two 'characters'). Paragraphs 2, 3, and 4 total 308 characters. Thus these two segments are approximately the same length. Now, one of the most distinctive differences between modern and classical Chinese is the use of *de* 的 (telecode 4104) in modern Chinese versus *zhi* 之 (telecode 0037) in classical Chinese (although the latter often appears in some styles of modern written Chinese, as is true in the case discussed here). Both *de* and *zhi* have the same syntactic function when they occur before nouns, i.e., relativizer or possessive marker. These properties are too well-known in the linguistic literature to require elaboration here.

In this excerpt, the *de*'s and *zhi*'s which have the same syntactic function are circled. Those which are lexicalized or which are not mutually substitutible (e.g. zhi preceding a monosyllabic noun) are underscored, and are not relevant to our discussion. For the circled characters, paragraph 1 ends up with a count of 11 de's and 1 zhi; whereas for paragraphs 2, 3, and 4 together there are only 5 de's but 16 zhi's. Thus in the first paragraph there is a high degree of consistency in the use of the modern literary de. In the next three paragraphs there is a greater mixture, but the bias is definitely towards use of the classical zhi.

How did such a state of affairs come about? Is the writer so highly inconsistent in his choice of vocabulary that he switches unconsciously back and forth from one style to the other? Note that there is a great deal more consistency rather than inconsistency when one considers the text in terms of two separate sections. It is our conjecture that the second section was written by a different author and then incorporated into the full text. Since in scientific writing it is quite common for an article to have multiple authorship, this type of stylistic alternation is by no means unusual, although in most instances the stylistic contrast is not as dramatic as it is in the example presented here. In any event, this is another problem which a machine translation system must deal with when it attempts to work on a language such as Chinese, which has long literary traditions of various styles of writing which depart from the spoken language. It is as if in English one were to come upon a paragraph of modern written English followed by another paragraph of Shakespearean or even Anglo-Saxon English. It is futile to try to write one grammar which will successfully analyze such a mixture of styles. A clear conception of what constitutes different styles of writing and the appropriate grammars which go with them must be formulated, and the grammars must be interfaced with great care. Our approach of utilizing different subgrammars which interact with one

paragraph 1

另一種藉物質消失轉變爲能的原子核反應稱爲「原子核分裂反應」。最常利用的是將一顆中子打入鈾-235的原子核中，由於新加入的中子使鈾原子核失去平衡狀態而發生原子核分裂。將「分裂產物」總加起來，其質量常少於鈾原子核加一個中子之總和，因此顯然在原子核分裂過程中有物質消失，經計算結果，每一顆鈾-235原子核加一顆中子後，經過核分裂反應可以因物質的消滅產生近兩億電子伏特的能量。(電子伏特爲能量單位之一)。若能將此項能量全部轉爲電能，只要千分之一克的鈾塊即可使一百瓦的電燈繼續點燃十二天半之久。核反應器中產生的熱，就是依靠鈾原子核分裂作用。由於核反應器及附屬機件也消耗能量，所以用反應器的熱產生蒸汽推動渦輪發生電能，效率在目前階段與燃煤發電相伯仲，約在百分之二十五左右，但正在不斷改進中。將來世界動力的廉價來源之中，原子能的利用將必爲其中之一。

1.3 原子能發展之需要

paragraph 2

原子能之被重視，除軍事上之用途而外，基於日常的需要。據研究全世界能量之消耗，因人口的增加，工業的進步，生活水準之提高，每年皆在快速增加之中。而在原子能發現以前世界能量之來源，則以採自化石燃料，如煤、石油爲主。但地殼內化石燃料之蘊藏量不足未來百年之需。是以開發能量之來源，乃爲當前迫不及待之舉，原子能既應運而生，綜合核分裂及融合能之應用，其燃料之蘊藏量足供今後數千萬年之需。

paragraph 3

其次，原子能應用於科學之研究及發展，可以幫助人類窺察宇宙萬物之奧秘；放射性同位素，應用於工業、農業及醫藥衛生等各方面，其範疇正在日漸擴大，效果亦在逐步增強，對於人類幸福生活之促進，亦有不容忽視之貢獻。

paragraph 4

綜而言之，原子能對於人類的前程，科學的發展及國家之進步，均有密切深厚之關聯，故其發展，極爲必需。

from *Yuan-zi Neng zhi Ji-ben Li-lun yu Yuan-zi Neng zhi He-ping Gong-xian* (Fundamental Theory of Atomic Energy and Peaceful Contributions of Atomic Energy)

another so as to be able to incorporate or interpret information involving possible differences in style is a first attempt to deal with this situation in a concrete way.

MISSION
of
Rome Air Development Center

RADC is the principal AFSC organization charged with planning and executing the USAF exploratory and advanced development programs for information sciences, intelligence, command, control and communications technology, products and services oriented to the needs of the USAF. Primary RADC mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, and electronic reliability, maintainability and compatibility. RADC has mission responsibility as assigned by AFSC for demonstration and acquisition of selected subsystems and systems in the intelligence, mapping, charting, command, control and communications areas.